



TECHNISCHE UNIVERSITÄT ILMENAU

Fakultät für Informatik und Automatisierung

Institut für Theoretische und Technische Informatik

Fachgebiet Integrierte Hard- und Softwaresysteme

# Diplomarbeit

zur Erlangung des akademischen Grades Diplom-Informatiker

## Vernetzung internetbasierter Portaldienste

---

<b>Inventarisierungsnummer:</b>	2007-10-02 / 141 / IN02 / 2235
<b>Autor:</b>	Christian Saul
<b>Geboren am:</b>	13.08.1983
<b>Matrikelnummer:</b>	35214
<b>Verantwortlicher Hochschulprofessor:</b>	Prof. Dr.-Ing. Andreas Mitschele-Thiel
<b>Wissenschaftlicher Betreuer:</b>	Dr.-Ing. Heinz-Dietrich Wuttke Dipl.-Ing. Ali Diab

Eingereicht am: 02.10.2007

# Vorwort

*Das Problem zu erkennen ist wichtiger, als die Lösung zu erkennen,  
denn die genaue Darstellung des Problems führt zur Lösung.*

Albert Einstein

Hiermit möchte ich denjenigen danken, die mich während der Erstellung dieser Diplomarbeit unterstützt und so auf unterschiedliche Weise zu ihrer Fertigstellung beigetragen haben.

Zunächst danke ich meinen Betreuern Dr.-Ing. Heinz-Dietrich Wuttke und Dipl.-Ing Ali Diab für die mir entgegengebrachte Unterstützung und die gute Zusammenarbeit.

Ein besonderer Dank gilt auch Prof. Dr.-Ing. Andreas Mitschele-Thiel für das Vertrauen als verantwortlicher Hochschulprofessor.

Der größte Dank gilt meinen Eltern, die mich während meines gesamten Studiums unterstützt und auf diese Weise die Erstellung der Diplomarbeit erst ermöglicht haben.

Abschließend möchte ich mich bei meiner Freundin für die vielen, mit Geduld ertragenen Entbehrungen während meines Studiums herzlich bedanken.

Christian Saul

# Kurzfassung

Das Internet als weltgrößtes Computernetzwerk enthält eine unvorstellbare Menge an global verfügbaren Informationen. Die Fülle an Informationen erreichte eine Komplexität, die viele Nutzer überforderte und eine zentrale und übersichtliche Strukturierung und Bündelung erforderlich machte. Aus dieser Notwendigkeit heraus entstanden sogenannte Portale als personalisierter Zugang zu Informationen und Diensten.

Portale können nur dann erfolgreich sein, wenn sie den Nutzern ein attraktives Angebot an Informationen und Diensten bieten. Damit ein Portal seinen Nutzern eine Fülle an Informationen und Diensten zur Verfügung stellen kann, ist Integration notwendig. Gegenstand dieser Arbeit ist die Integration von Diensten des Bildungsportals Thüringen, der Bibliothek und der Alumni-Datenbank der Technischen Universität Ilmenau in das Alumni-Weiterbildungswbportal. Damit diese Dienste integriert werden können, müssen Schnittstellen konzipiert und geeignete Integrationstechnologien ausgewählt werden.

Das Bildungsportal Thüringen besitzt bereits eine Schnittstelle für die Integration. Diese eignet sich hervorragend für eine Integration und macht eine Neu-Konzeption unnötig. Im Gegensatz dazu besitzen die zwei weiteren Dienste noch keine Schnittstellen.

Die Analyse der Dienste zeigt, dass die Schnittstelle zur Bibliothek durch eine Message-orientierte Middleware und die Schnittstelle zur Alumni-Datenbank durch einen Web Service realisiert werden soll. Nach der Konzeption werden diese Schnittstellen prototypisch implementiert.

# Abstract

The internet as the world's largest computer network contains an incredibly large amount of global available information. This gained a complexity that overstrained many users and required central and clear structuring and grouping. Out of this need portals emerged, which are enabling users a personalized access to information and services.

Portals are only successful if they provide an attractive offer of information and services. In order to make information and services available to users, they need to be integrated. The focus of this work is the integration of services from Bildungsportal Thüringen, from the library and the Alumni-database of the TU Ilmenau into the Alumni-Weiterbildungswbportal. To integrate these services interfaces have to be designed and an appropriate integration-technology has to be selected.

The Bildungsportal Thüringen already has an interface for integration. This interface is particularly suitable for integration and makes a complete redesign unnecessary. In contrast, the two additional services do not possess an interface.

Analyses show that the interface to the library shall be realized by a message-oriented-middleware and the interface to the Alumni-database by a web-service. After the conceptual design the interfaces are implemented prototypical.

# Inhaltsverzeichnis

ABBILDUNGSVERZEICHNIS .....	V
TABELLENVERZEICHNIS .....	VII
LISTING-VERZEICHNIS .....	VII
1. EINLEITUNG .....	1
1. 1. Motivation .....	2
1. 2. Aufgabenstellung .....	2
1. 3. Aufbau der Arbeit .....	3
2. STAND DER TECHNIK .....	5
2. 1. Portale .....	5
2. 1. 1. Definition .....	5
2. 1. 2. Das ideale Portal .....	6
2. 1. 3. Eigenschaften eines Portals .....	6
2. 1. 4. Portal-Architekturmodell .....	8
2. 1. 5. Portal-Referenzarchitektur .....	10
2. 1. 6. Integration von Anwendungen in Portale .....	15
2. 1. 7. Klassifizierung von Portalen .....	17
2. 1. 8. Abgrenzung des Portalbegriffs .....	19
2. 1. 9. Zusammenfassung .....	20
2. 2. Web-Technologien .....	21
2. 2. 1. Basistechnologien .....	21
2. 2. 2. Scriptsprachen .....	23
2. 2. 3. XML .....	24
2. 2. 4. Web 2.0 .....	28
2. 2. 5. SOA .....	32
2. 2. 6. Web Services .....	35
2. 3. Integration von Anwendungen .....	43
2. 3. 1. Integrationstopologien .....	43
2. 3. 2. Integrationsebenen .....	45
2. 3. 3. Differenzierungsmerkmale .....	53
2. 3. 4. Integrationstechnologien .....	56
2. 3. 5. Zusammenfassung .....	59

3.	DAS ILMENAUER ALUMNI-PORTAL.....	61
3. 1.	Alumni, Alumni-Vereinigung .....	61
3. 2.	Alumni-Arbeit .....	62
3. 2. 1.	Alumni-Arbeit in den USA.....	62
3. 2. 2.	Alumni-Arbeit in Deutschland .....	62
3. 3.	Alumni-Netzwerke .....	64
3. 3. 1.	Absolventen-Netzwerk der TU Ilmenau.....	64
3. 3. 2.	Absolventen-Netzwerke anderer Hochschulen .....	65
3. 4.	Alumni-Portale.....	65
3. 5.	Konzeption des Ilmenauer Alumni-Portals .....	67
3. 5. 1.	Handlungsempfehlungen zukünftiger Alumni-Arbeit .....	67
3. 5. 2.	Das Projekt AWWP.....	68
3. 5. 3.	Herausstellungsmerkmale zu anderen Alumni-Portalen.....	71
3. 6.	Realisierung des Ilmenauer Alumni-Portals.....	73
3. 6. 1.	Portal-Software-Lösungen.....	73
3. 6. 2.	Unternehmens-Portale.....	74
3. 6. 3.	Make-or-Buy .....	75
3. 6. 4.	Die eigene Portal-Lösung.....	77
3. 6. 5.	Alumni-Portal-Lösung.....	78
4.	ANALYSE DER DIENSTE DES ALUMNI-PORTALS.....	80
4. 1.	Dienste des Alumni-Portals .....	80
4. 1. 1.	Lebenslauf.....	80
4. 1. 2.	Veröffentlichungen .....	80
4. 1. 3.	Ilmedia .....	81
4. 1. 4.	News & Events .....	81
4. 1. 5.	Alumni-News.....	81
4. 1. 6.	Career-Service.....	81
4. 1. 7.	Jobbörse.....	81
4. 1. 8.	Weiterbildung .....	81
4. 1. 9.	E-Learning .....	82
4. 1. 10.	Bildungsportal Thüringen .....	82
4. 1. 11.	Präsentation der TU Ilmenau .....	82
4. 1. 12.	Administrator-Funktionalität.....	82
4. 1. 13.	Nutzerkonto.....	83
4. 1. 14.	AlumniDB .....	83

4. 2.	Klassifikationskriterien .....	84
4. 2. 1.	Integrationslokation .....	84
4. 2. 2.	Informationsfluss .....	84
4. 2. 3.	Kommunikationsmodell .....	86
4. 2. 4.	Dienstursprung .....	86
4. 3.	Analyse .....	88
4. 3. 1.	Überblick .....	88
4. 3. 2.	Analyse im Detail .....	91
4. 4.	Zusammenfassung .....	91
5.	KONZEPTION DER SCHNITTSTELLEN .....	93
5. 1.	Bildungsportal Thüringen .....	94
5. 1. 1.	Allgemeines zum BPT-Dienst .....	94
5. 1. 2.	Nutzen des BPT-Dienstes .....	94
5. 1. 3.	Analyse der BPT-Schnittstelle .....	95
5. 1. 4.	Eignung für das Alumni-Portal .....	99
5. 2.	Ilmedia .....	101
5. 2. 1.	Allgemeines zum Ilmedia-Dienst .....	101
5. 2. 2.	Nutzen des Ilmedia-Dienstes .....	102
5. 2. 3.	Konzeption der Ilmedia-Schnittstelle .....	102
5. 2. 4.	Zusammenfassung .....	107
5. 3.	Alumni-Datenbank .....	108
5. 3. 1.	Allgemeines zur Alumni-Datenbank .....	108
5. 3. 2.	Nutzen der Alumni-Datenbank .....	108
5. 3. 3.	Konzeption der AlumniDB-Schnittstelle .....	108
5. 3. 4.	Zusammenfassung .....	111
6.	IMPLEMENTIERUNG DER SCHNITTSTELLEN .....	113
6. 1.	Ilmedia-Schnittstelle .....	113
6. 1. 1.	Entwicklung der Ilmedia-XSD .....	113
6. 1. 2.	Implementierung in PHP .....	121
6. 1. 3.	Entwicklung des Ilmedia-XSLT-Stylesheets .....	130
6. 2.	AlumniDB-Schnittstelle .....	134
6. 2. 1.	Entwicklung der AlumniDB-WSDL .....	134
6. 2. 2.	Implementierung des Web Services in Java .....	138
6. 2. 3.	Implementierung des Web Services in PHP .....	148

7.	ZUSAMMENFASSUNG UND AUSBLICK.....	153
7. 1.	Zusammenfassung .....	153
7. 2.	Ausblick.....	154
ANHANG.....		155
Anhang A: Analyse der Dienste im Detail .....		155
LITERATURVERZEICHNIS.....		173
THESEN ZUR DIPLOMARBEIT .....		178
EIDESSTATTLICHE ERKLÄRUNG .....		180



# Abbildungsverzeichnis

Abbildung 1: Portal-Architekturmodell (Quelle: [4]).....	1
Abbildung 2: Portal-Referenzarchitektur (Quelle: [5]).....	1
Abbildung 3: Portlets .....	1
Abbildung 4: Rendering einer Portal-Seite (Quelle: [40]).....	1
Abbildung 5: Single-Sign-On (Quelle: [40]) .....	1
Abbildung 6: Anwendungsorientierte Portal-Integration (Quelle: [40]).....	1
Abbildung 7: Datenintegration in das Portal (Quelle: [40]).....	1
Abbildung 8: Grundkomponenten des Web .....	1
Abbildung 9: Schema der Transformation (Quelle: [43]) .....	1
Abbildung 10: XML - Programmierschnittstellen .....	1
Abbildung 11: SOA- Tempel (Quelle: [47]).....	1
Abbildung 12: Rollenmodell der SOA (Quelle: [16]).....	1
Abbildung 13: Web-Service-Rollenmodell (Quelle: [16]) .....	1
Abbildung 14: Web-Service-Szenario (Quelle: [19]).....	1
Abbildung 15: Grundlegender Aufbau einer SOAP-Message (Quelle: [16]) .....	1
Abbildung 16: Aufbau einer WSDL-Beschreibung (Quelle: [20]) .....	1
Abbildung 17: Punkt-zu-Punkt-Integration .....	1
Abbildung 18: Bus-Topologie .....	1
Abbildung 19: Hub-and-Spoke-Topologie.....	1
Abbildung 20: Aufbau einer Anwendung (MVC-Model) .....	1
Abbildung 21: Transformationen bei der Datenintegration.....	1
Abbildung 22: Datenintegration .....	1
Abbildung 23: Funktionsintegration .....	1
Abbildung 24: Benutzerschnittstellenintegration .....	1
Abbildung 25: API-basierter Dienstaufbau.....	1
Abbildung 26: Nachrichten-basierter Dienstaufbau.....	1
Abbildung 27: Einordnung der Integrationstechnologien .....	1
Abbildung 28: Zusammenspiel der Dienste des Alumni-Portals.....	1
Abbildung 29: Markteinteilung nach Gartner (Quelle: [33]) .....	1
Abbildung 30: Unternehmensportale (Quelle: [40]) .....	1
Abbildung 31: Informationsfluss im Web 1.0 .....	1
Abbildung 32: Informationsfluss im Web 2.0 .....	1
Abbildung 33: Dienstsprung.....	1
Abbildung 34: Schnittstelle zum Bildungsportal Thüringen .....	1
Abbildung 35: Ilmedia (Quelle: [35]) .....	1
Abbildung 36: Schnittstelle zur Bibliothek der TU Ilmenau.....	1
Abbildung 37: Schnittstelle zur Alumni-DB.....	1
Abbildung 38: AlumniDB-Web-Service .....	1
Abbildung 39: Hierarchie der Ilmedia-XSD .....	1
Abbildung 40: Graphische Darstellung der Ilmedia-XSD in XMLSpy .....	120
Abbildung 41: Daten der zu exportierenden Publikation.....	1
Abbildung 42: Darstellung der publication.xml gemäß dem XSLT-Stylesheet .....	133
Abbildung 43: Erstellen einer Web-Anwendung in NetBeans.....	140
Abbildung 44: Web-Service-Tester des SJSAS 9 .....	142
Abbildung 45: SOAP-Request .....	143
Abbildung 46: SOAP-Response .....	143

Abbildung 47: Web-Service-Client in Java.....	146
Abbildung 48: Der Dienst Lebenslauf.....	156
Abbildung 49: Der Dienst Veröffentlichungen .....	158
Abbildung 50: Der Dienst News & Events .....	160
Abbildung 51: Der Dienst Alumni-News.....	161
Abbildung 52: Der Dienst Career-Service.....	163
Abbildung 53: Der Dienst Jöbbörse.....	164
Abbildung 54: Der Dienst Weiterbildung .....	165
Abbildung 55: Der Dienst Administrator-Funktionalität .....	169
Abbildung 56: Der Dienst Nutzerkonto .....	170

# Tabellenverzeichnis

Tabelle 1: Vor- und Nachteile der Programmiermodelle DOM und SAX .....	27
Tabelle 2: WSDL - Operationen .....	40
Tabelle 3: Vor- und Nachteile der Datenintegration .....	49
Tabelle 4: Vor- und Nachteile der Funktionsintegration .....	50
Tabelle 5: Vor- und Nachteile der Benutzerschnittstellenintegration .....	52
Tabelle 6: Die wichtigsten Klassen des DOM in PHP 5 (Quelle: [37]) .....	123

# Listing-Verzeichnis

Listing 1: Nachricht in XML-Format .....	55
Listing 2: XML-Dokument vom BPT .....	98
Listing 3: publications-Element .....	116
Listing 4: publication-Element .....	116
Listing 5: comment-Element .....	116
Listing 6: authors-Element .....	117
Listing 7: persons-Element .....	117
Listing 8: publication_informations-Element .....	118
Listing 9: Einfache Elemente .....	119
Listing 10: Die Klasse Publication .....	124
Listing 11: Die Klasse Publications .....	125
Listing 12: Beispielcode zur Erzeugung einer Publikation in XML .....	128
Listing 13: publication.xml .....	129
Listing 14: Verarbeitungsanweisung in PHP .....	130
Listing 15: Verarbeitungsanweisung in XML .....	130
Listing 16: Imedia-XSLT-Stylesheet .....	131
Listing 17: XSL-Transformation des authors-Elements .....	132
Listing 18: definitions-Element .....	135
Listing 19: message-Element .....	135
Listing 20: portType-Element .....	136
Listing 21: binding-Element .....	137
Listing 22: service-Element .....	137
Listing 23: AlumniDB-Web-Service in Java (ohne WSDL) .....	141
Listing 24: AlumniDB Web Service in Java (mit WSDL) .....	145
Listing 25: AlumniDB-Web-Service-Client in Java .....	147
Listing 26: php.ini .....	148
Listing 27: AlumniDB Web Service in PHP (ohne WSDL) .....	149
Listing 28: AlumniDB-Web-Service in PHP (mit WSDL) .....	150
Listing 29: Web-Service-Client in PHP (mit WSDL) .....	151
Listing 30: Web-Service-Client in PHP (ohne WSDL) .....	152
Listing 31: Abfragen von Web-Service-Informationen .....	152
Listing 32: Ergebnis der Web-Service-Abfrage .....	152



# 1. Einleitung

Im Jahre 1969 begann das amerikanische Verteidigungsministerium damit, das erste große Computernetzwerk aufzubauen. Das sogenannte ARPA-Netz wurde zur Vernetzung von Universitäten und Forschungseinrichtungen benutzt, mit dem Ziel Verbindungen zwischen einzelnen Computern herzustellen und so Daten auszutauschen. Daraus entwickelten sich im Laufe der Jahre das Internet und das allseits bekannte World Wide Web<sup>1</sup>. Doch mit der Verbreitung und Nutzung des Internets nahm das Angebot an global verfügbaren Informationen zu. Die Fülle an Informationen erreichte jedoch schnell eine Komplexität, die viele Nutzer überforderte und eine zentrale und übersichtliche Strukturierung und Bündelung erforderlich machte. Aus dieser Notwendigkeit heraus entstanden sogenannte Portale als Einstiegsseiten in das Internet.

Der Begriff Portal als Bezeichnung für eine Web-Seite entstand erst im Jahre 1998. Portale erleichterten den Nutzern die Orientierung und Navigation in der Informationsflut des Internets, durch Themenbereiche und Suchfunktionen. Durch das weiterhin exponentielle Wachstum an Daten und Informationen im Internet reichte den Nutzern die von ursprünglichen Portalen angebotene Navigationsunterstützung nicht mehr aus. Weitere Funktionalitäten, wie die Personalisierung der Seiten, wurden in Portale integriert.

Durch die Möglichkeit der spezifischen und anforderungsgerechten Anpassung der Portale durch den Nutzer und die Integration einzelner Systeme und Anwendungen unterscheidet sich das heutige Portal weitgehend von den ursprünglichen Web-Seiten, deren Sinn und Zweck vorwiegend in der Werbe- und Repräsentationsaufgabe bestand.

---

<sup>1</sup> Die Begriffe World Wide Web, Web und WWW werden im Folgenden synonym verwendet.

## 1. 1. Motivation

Portale können nur dann erfolgreich sein, wenn sie den Nutzern ein attraktives Angebot an Informationen und Diensten bieten. Nur wenn die Nutzer das Portal als Bereicherung ansehen, werden sie es nutzen und eigene Erfahrungen und eigenes Wissen einbringen.

Damit ein Portal seinen Nutzern eine Fülle an Informationen und Diensten zur Verfügung stellen kann ist Integration notwendig. Die Integration bezeichnet die Prozessabbildung und den Datenaustausch zwischen verschiedenen heterogenen Anwendungen und stellt eine Grundlage für Portale dar. Ein Portal muss eine Infrastruktur anbieten, die es ermöglicht, andere Dienste einzubinden. Dabei darf es keine Rolle spielen, ob diese Anwendungen an unterschiedlichen Standorten betrieben werden.

## 1. 2. Aufgabenstellung

Gegenstand dieser Arbeit ist die Vernetzung des Alumni-Weiterbildungswebportals (AWWP) mit Diensten des Bildungsportals Thüringen, der Bibliothek (Ilmedia) und der Alumni Datenbank der TU Ilmenau. Dieses Portal wird im Rahmen eines Projektes im Fachgebiet Integrierte Hard- und Software-Systeme der Fakultät für Informatik und Automatisierung entwickelt.

Zunächst sind Methoden und Standardtechnologien für die Integration von Anwendungen zu untersuchen und den Stand der Technik zu erfassen. Darauf aufbauend sind Schnittstellen zu spezifizieren, die sich für die Integration der Dienste des Bildungsportals Thüringen, der Bibliothek und der Alumni Datenbank der TU Ilmenau eignen. Die Ergebnisse sollen in das Projekt AWWP einfließen. Exemplarisch sind diese zu implementieren.

## 1. 3.      **Aufbau der Arbeit**

Der Aufbau der Arbeit gliedert sich wie folgt:

Das zweite Kapitel der Arbeit beschäftigt sich mit der allgemeinen Definition von Portalen in der Informatik sowie deren Abgrenzung gegen andere Lösungen der vernetzten Welt. Anschließend werden die einzelnen Technologien betrachtet und erläutert, welche Grundlagen für Portale sind. Im letzten Teil des Kapitels werden konkrete Integrationstechnologien vorgestellt und bewertet, die für die Integration von heterogenen Anwendungen grundlegend sind.

Das dritte Kapitel gibt einen Aufschluss über das neue Ilmenauer Alumni-Portal. Dazu werden zuerst die Begriffe Alumni, Alumni-Vereinigung und Alumni-Arbeit geklärt, bevor konkrete Alumni-Netzwerke betrachtet werden. Anschließend wird erläutert, warum sich Portale hervorragend für die Alumni-Arbeit eignen. Nach diesen grundsätzlichen Betrachtungen rückt das neue Ilmenauer Alumni-Portal in den Mittelpunkt. Dabei wird verdeutlicht, aus welcher Situation es entstanden ist und wodurch es sich von anderen Absolventen-Portalen abhebt. Nach den konzeptionellen Darstellungen wird die konkrete Umsetzung des Portals beleuchtet. Eine Erörterung, warum das neue Alumni-Portal nicht durch eine Standard-Portal-Lösung realisiert werden sollte, schließt das Kapitel ab.

Das vierte Kapitel analysiert die Dienste des neuen Ilmenauer Alumni-Portals. Dazu werden zunächst Kriterien erarbeitet, nach denen die Dienste klassifiziert werden.

Das fünfte Kapitel befasst sich mit der Analyse der Dienste des Bildungsportals Thüringen, der Bibliothek sowie der Alumni-Datenbank der TU Ilmenau. Darauf aufbauend werden Schnittstellen konzipiert, um die drei Dienste in das Alumni-Portal zu integrieren.

Das sechste Kapitel implementiert die im fünften Kapitel entworfenen Schnittstellen.

Das letzte Kapitel fasst die Arbeit zusammen und schließt mit einem Ausblick ab.





## 2. Stand der Technik

Dieses Kapitel definiert die für die vorliegende Diplomarbeit benötigten fachlichen Grundlagen. Dadurch wird ein gemeinsames Begriffsverständnis für die nachfolgenden Kapitel geschaffen.

### 2. 1. Portale

#### 2. 1. 1. Definition

Das Wort *Portal* hat seinen Ursprung in dem lateinischen Wort *porta* und bedeutet soviel wie *Pforte*. In der Baukunst bezeichnet die Pforte einen architektonisch besonders ausgestalteten Eingang zu einem größeren Gebäude, wie etwa zu einem Schloss oder einer Kirche. Während sich die architektonische Gestaltung von Portalen von der Antike, über die Romantik, Gotik, Renaissance und Barock bis in die Neuzeit wandelte, blieb die wesentliche Funktion der Portale im Laufe der Zeit konstant, nämlich einen eindrucksvollen Eingang zu einer dahinter liegenden, räumlich abgegrenzten und inhaltlich zusammenhängenden Bereich zu kennzeichnen.

Um die Bedeutung des Begriffs Portal in der Sprache der Informatik zu klären, bedarf es mehr Mühe, denn der Portal-Hype der letzten Jahre führte zu einer Verwässerung des Begriffs. Ein Grund dafür war die anfängliche Euphorie über die Lösung aller Probleme durch Einsatz eines Portals. Allgemeiner als ein Portal als Eintrittspunkt zu einem Informationssystem, hätte eine Definition nicht ausfallen können. Erschwerend kam hinzu, dass viele Produkte und Lösungen als Portal-Lösungen verkauft wurden.

Nachfolgende Definition versucht die vage Vorstellung eines Portals etwas genauer zu fassen:

*„Ein Portal ist definiert als eine Applikation, welche basierend auf Webtechnologien einen zentralen Zugriff auf personalisierte Inhalte sowie bedarfsgerecht auf Prozesse bereitstellt. Charakterisierend für Portale ist die Verknüpfung und der Datenaustausch zwischen heterogenen Anwendungen über eine Portalplattform. Eine manuelle Anmeldung an den in*

*das Portal integrierten Anwendungen ist durch Single-Sign-On nicht mehr notwendig, es gibt einen zentralen Zugriff über eine homogene Benutzungsoberfläche. Portale bieten die Möglichkeit, Prozesse und Zusammenarbeit innerhalb heterogener Gruppen zu unterstützen.“ [1]*

Bei einem Portal steht das Bereitstellen von applikationsübergreifenden Leistungen und somit der Integrationsaspekt und nicht die technische Implementierung im Vordergrund. Ein Portal muss nicht auf Webtechnologien basieren, auch wenn es oftmals der Fall ist.

### **2. 1. 2. Das ideale Portal**

Die Firma Dataquest, ein Marktforschungsunternehmen definierte das *ideale Portal* wie folgt: „*Das ideale Portal eröffnet einen gemeinsamen, personalisierten Zugang zu Daten, Expertisen und Anwendungen*“ [2]. In diesem Zusammenhang soll deutlich werden, dass genannte Funktionen nicht bindend und kaum alle in einem Portal zu finden sind. Lediglich ein ideales Portal erfüllt diese Kriterien. Dieser idealisierte Zusammenhang ist nützlich für den Vergleich zwischen Portalen und herkömmlichen Web-Seiten.

### **2. 1. 3. Eigenschaften eines Portals**

Eine Web-Seite darf als Portal bezeichnet werden, wenn sie durch folgende Eigenschaften gekennzeichnet ist. [3]

#### *Search and Navigation*

Eine Navigation erleichtert die Suche nach Inhalten. Dazu gehören zum einen die Oberflächensuche, also die Suche nach dem angegebenen Dateinamen und zum anderen auch die Suche nach den Inhalten der Dateien. Suchmaschinen in Portalen suchen in der Regel nicht das gesamte WWW ab, sondern nur die Informationsbestände und Anwendungssysteme, die mit Hilfe des Portals zugänglich sind.

*Information-Integration (Content-Management)*

Da die Navigation Informationen aus verschiedenen Quellen bereitstellen soll, erfordert dies eine Integration von verschiedenen Inhalten und wird unter dem Begriff Content-Management zusammengefasst. Unter Content-Management versteht man nicht nur die Bereitstellung von Inhalten aus verschiedenen Quellen, sondern auch deren effiziente Nutzung.

*Personalization*

Der Informationsüberfluss ist eines der größten Probleme des Internets. Das Web bietet eine fast unendliche Menge an Daten. Eine effiziente Verwendung von Informationen ist nur möglich, wenn eine Vorauswahl stattgefunden hat. Diese Filterfunktion übernimmt die Personalisierung. Die angebotenen Daten und Dienste können auf diese Weise individuell an die Gewohnheiten und Vorlieben eines Nutzers angepasst werden. Dabei wird der jeweilige Portal-Nutzer zum Mittelpunkt des Geschehens.

*Notification (Push-Technology)*

Ein Portal soll zudem in der Lage sein, dem Nutzer automatisiert Informationen in Form von Nachrichten zukommen zu lassen, ohne dass der Nutzer sie explizit anfordert.

*Collaboration and Groupware*

Diese bisher beschriebenen Funktionen ergeben nur in einem Umfeld geeigneter Informationen einen Sinn. Damit diese am zweckmäßigen Ort und zur richtigen Zeit vorhanden sind, ist die fünfte Funktion der Portale notwendig, die als Knowledge-Management bezeichnet wird. Inner- und außerhalb eines Portals wird täglich für den Portalnutzer relevantes Wissen erstellt. Durch die passende Strukturierung und individuelle Zuführung dieses Wissens wird ein kosten- und innovationsrelevanter Vorteil geschaffen. Durch solche Wissensnetzwerke gestaltet sich die Suche sowohl einfacher als auch effektiver.

*Task-Management and Workflow*

Ein weiterer Effizienzvorteil entsteht durch eine geeignete Arbeitsablaufgestaltung, dem sogenannten Workflow-Management. Dadurch wird asynchrone, das bedeutet nicht aufeinander abgestimmte, arbeitsteilige Gruppenarbeit koordiniert.

### *Integration of Applications and Business-Intelligence*

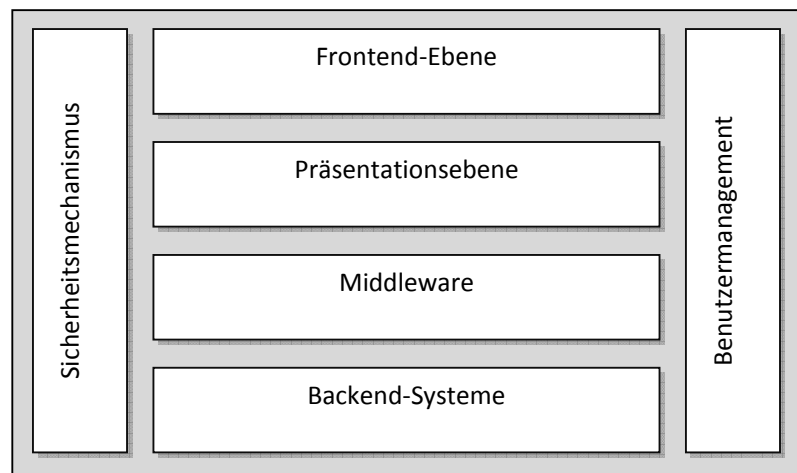
Die Integration von Anwendungen ist die siebte Funktion eines idealen Portals. Man unterscheidet zwischen der daten- und anwendungsorientierten Integration. Da die Integration von Anwendungen für die weiteren Betrachtungen von großer Bedeutung ist, wird sie in dem Kapitel 2.1.6 noch einmal genauer betrachtet.

### *Infrastructure-Functionality*

Die zuletzt genannte Funktion ist die Basis, auf der alle anderen Funktionen aufbauen. Diese als Infrastrukturdienst bezeichneten Dienste machen andere erst möglich. Als Basisfunktion lässt sie alle oben genannten Funktionen zu einer Einheit werden, von der aus alle anderen verfügbar sind.

## **2. 1. 4. Portal-Architekturmodell**

Technisch gesehen ist ein Portal kein isoliertes System, sondern bündelt bestehende Anwendungen und Informationsdienste. Sein Aufbau ist daher nicht monolithisch, sondern kann grob mit sechs Komponenten beschrieben werden. [4]



**Abbildung 1: Portal-Architekturmodell (Quelle: [4])**

### *Frontend*

Diese Komponente enthält alle Frontends, über die ein Portal erreicht werden kann. Als Frontend wird eine Applikation bezeichnet, die zur interaktiven Anforderung, Eingabe sowie Anzeige von Daten verwendet wird. Die verwendeten Frontends sollten überall verfügbar sein. Web-Browser eignen sich hervorragend als Frontend, da sie integraler Bestandteil der meisten Betriebssysteme sind. Wichtig ist auch, dass das Frontend für den Anwender komfortabel ist, damit es ohne großen Schulungsaufwand bedient werden kann.

### *Präsentationsebene*

Die zweite Komponente eines Portals ist die Präsentationsebene. Sie bereitet die Daten aus der Middleware auf, sodass sie dem Benutzer mittels unterschiedlicher Frontends präsentiert werden können.

### *Middleware*

Die Middleware verbindet die Präsentationsebene mit den Backend-Systemen. Sie fasst Daten zusammen, kann Backend-Systeme anschließen und Daten aus unterschiedlichen Systemen verarbeiten. Über diese Komponente werden Daten aus verschiedenen Backend-Systemen untereinander ausgetauscht.

### *Backend-Systeme*

Das Backend ist bei allen Portalprojekten vorhanden und stellt in den meisten Fällen die Basis dar. Die Backend-Systeme repräsentieren die unterschiedlichen Informationsquellen eines Portals. Sie setzen sich unter anderem aus Anwendungen, Services oder Datenbanken zusammen.

### *Sicherheitsmechanismus/-management*

Diese Komponente beinhaltet alle technischen Einrichtungen sowie Regularien, die zur Sicherung des Portals nach innen und außen existieren. Der Sicherheitsmechanismus bildet zusammen mit dem Benutzermanagement den Rahmen des Portals.

### *Benutzermanagement*

Das Benutzermanagement besteht aus der Benutzer- und der Rollenverwaltung. Diese Komponente enthält alle technischen Einrichtungen, die sich mit den Benutzern eines Portals und dessen Rollen beschäftigen. Das Benutzermanagement ist neben dem Sicherheitsmechanismus der zweite Teil des Rahmens eines Portals.

## 2.1.5. Portal-Referenzarchitektur

Der Markt für Portal-Software ist schwer durchschaubar und es existieren viele unterschiedliche Ansätze im Bereich Technologie und verschieden ausgeprägte Funktionsumfänge der einzelnen Produkte. Aus der Vielfalt der Systeme wurde eine Referenzarchitektur abgeleitet, die den grundlegenden Aufbau und die Basisfunktionalitäten von Portal-Software aufzeigt. [5]

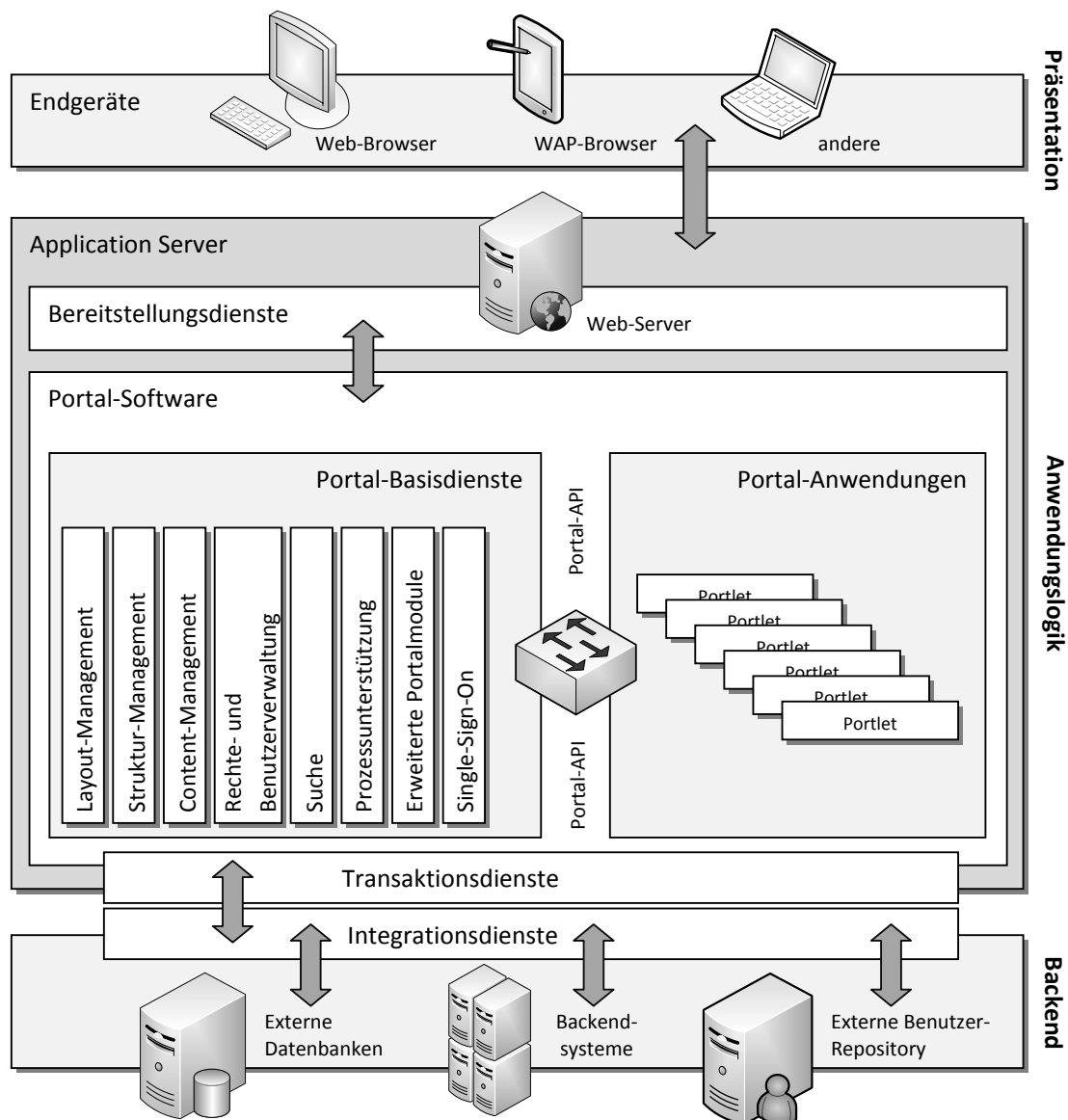


Abbildung 2: Portal-Referenzarchitektur (Quelle: [5])

Die Referenzarchitektur von Portalen besteht aus einer 3-Schichten-Architektur mit Präsentation, Anwendungslogik und Backend (vgl. Portal-Architekturmodell).

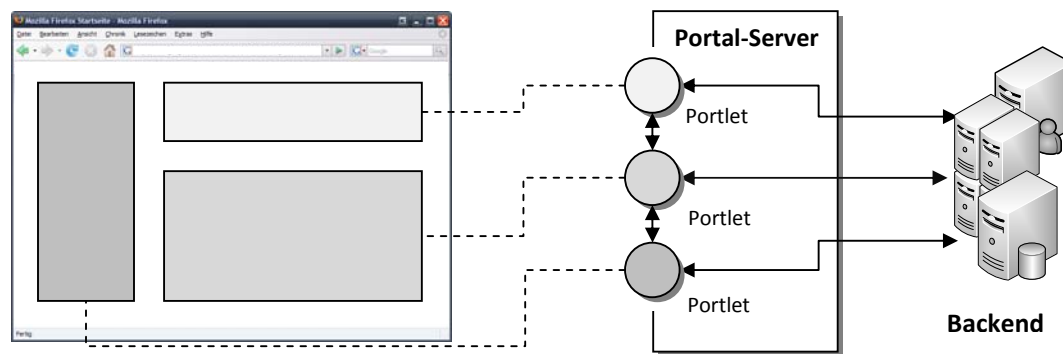
Der client-seitige Teil der Präsentationsschicht umfasst die Endgeräte der Portal-Nutzer. Die Schicht der Anwendungslogik besteht aus dem Bereitstellungsdienst, meist in Form eines Web-Servers, der die Anfragen der Clients entgegen nimmt und die Portal-Seiten ausliefert. Ebenfalls Teil der Anwendungslogik ist die eigentliche Portal-Software. Für den Betrieb der Portal-Software ist ein Application-Server erforderlich, der ein Framework für die Entwicklung von Anwendungen bereitstellt. Die Schnittstelle zur Backend-Schicht bilden die Integrationsdienste. Die Middleware-Technologien und Web Service unterstützenden Integrationsdienste, gestatten eine einfache Anbindung von bestehenden Systemen in das Portal. Weiterhin gewährleisten Transaktionsdienste die Transaktionssicherheit über die verschiedenen angebundenen Systeme hinweg.

Die Portal-Software bietet zur Realisierung von Portalen Basisdienste an, die von den spezifischen Portal-Anwendungen genutzt werden können. Portale besitzen auf diese Weise eine Framework-Funktion zur Realisierung von Anwendungen. Portal-Software besteht damit aus Portal-Basisdiensten und Portal-Anwendungen.

Die Portal-Anwendungen realisieren die Anwendungsfunktionalität des Portals und werden aus einer allgemeinen Portal-Anwendungsklasse instanziiert. Am Markt existieren Anwendungsklassen in verschiedenen Formen, wie zum Beispiel *Portlets*, *I-Views*, *I-Lets* und *Gadgets*. Portal-Anwendungen können über eine Softwareschnittstelle, der Portal-API, vom Portal aufgerufen werden und selbst über diese Schnittstelle Basisdienste des Portals nutzen. Die bekannteste Portal-API ist die *Portlet-API*<sup>1</sup>.

---

<sup>1</sup> JSR 168 Portlet Specification, Link: <http://www.jcp.org/en/jsr/detail?id=168>



**Abbildung 3: Portlets**

Die Basisdienste einer Portal-Software sind *Layout-, Struktur- und Content Management, Rechte- und Benutzerverwaltung, Suche, Prozessunterstützung* sowie *Single-Sign-On*.

### *Struktur-Management*

Das Struktur-Management definiert den strukturellen Aufbau und die Navigierbarkeit des Portals, wie es dem Nutzer präsentiert wird. Dies umfasst insbesondere die Definition, an welcher Stelle der Portal-Struktur Anwendungen platziert sind.

### *Layout-Management*

Die Aufgabe des Layout-Managements ist die Zusammenstellung der vom Nutzer angefragten Portal-Seiten aus den einzelnen Anwendungen und die Erzeugung der dem Endgerät des Nutzers entsprechenden spezifischen Ausgabe. Der Erstellungsprozess wird Rendering genannt. Bei der Erstellung werden die vom Portal-Betreiber und die vom Portal-Nutzer vorgegebenen Design-Vorlagen berücksichtigt. Das Rendering von Portal-Seiten besteht im Wesentlichen aus zwei Schritten. Im ersten Schritt werden die Daten aus den einzelnen Anwendungen und Systemen aggregiert, bevor im zweiten Schritt die einzelnen Seiten zusammengestellt, das Layout angepasst und Navigationselemente hinzugefügt werden.



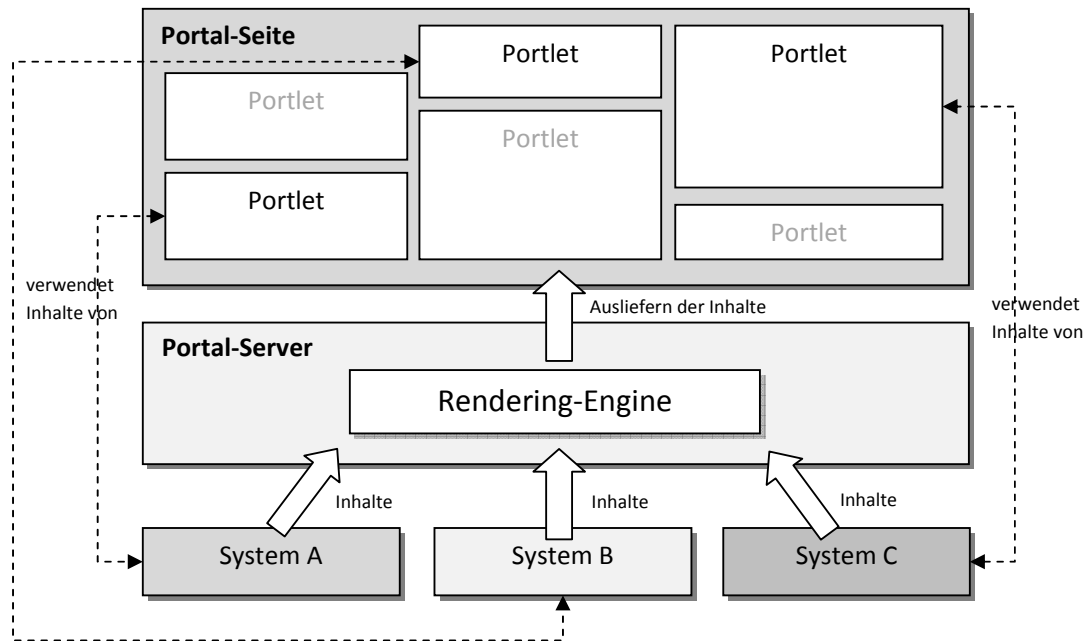


Abbildung 4: Rendering einer Portal-Seite (Quelle: [40])

### Suche

Die Suche über verschiedene, heterogene Datenbestände wird durch den Basisdienst Suche durchgeführt und ist eine zentrale Funktion in jedem Portal. Eine besondere Schwierigkeit liegt in der Darstellung der Suchergebnisse, da diese aus verschiedenen strukturierten Quellen und Anwendungen stammen.

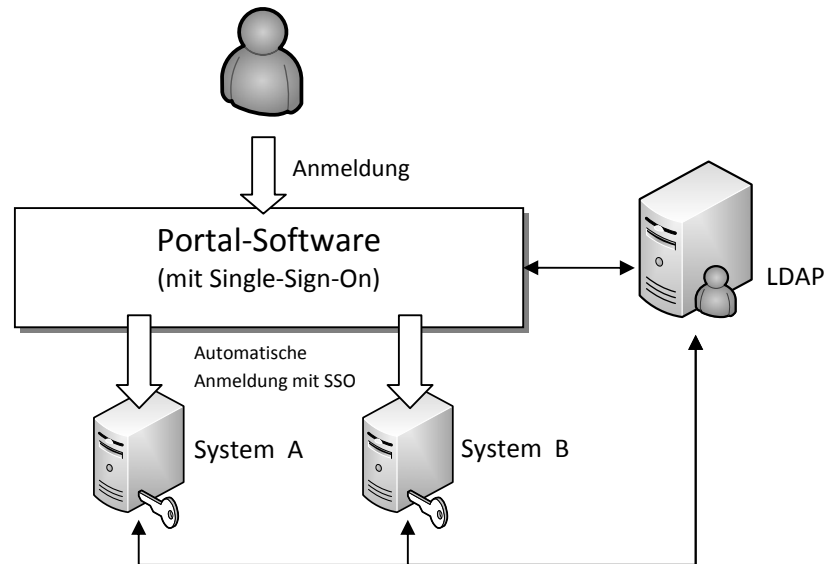
### Rechte- und Benutzerverwaltung

Die Rechte- und die Benutzerverwaltung ist ein weiterer Basisdienst der Portal-Software. Oftmals bieten die meisten Portale nur eine integrierte, rudimentäre Benutzerverwaltung. Aus diesem Grund wird häufig auf eine externe Benutzerverwaltung in Form eines unternehmensweiten Verzeichnisdienstes<sup>1</sup> zurückgegriffen.

### Single-Sign-On

Eng verbunden mit der Rechte- und Benutzerverwaltung ist der Single-Sign-On-Dienst. Der Dienst meldet die Nutzer des Portals automatisiert mit dem Passwort des Nutzers an allen in das Portal integrierten Systemen an.

<sup>1</sup> Weit verbreitet als Verzeichnisdienst ist LDAP (Lightweight Directory Access Protocol), <http://www.openldap.org/>



**Abbildung 5: Single-Sign-On (Quelle: [40])**

### *Content-Management*

Die Verwaltung von Inhalten ist bei einem Portal ebenso wichtig wie bei einem herkömmlichen Web-Auftritt. Auch in Portalen wird das Content-Management benötigt. Dieser Dienst verwaltet die Inhalte und stellt sie für das Portal bereit. Das Content-Management kann Bestandteil der Portal-Software sein oder als externes Content-Management-System integriert werden.

### *Erweiterte Portal-Module*

Zu den Diensten der erweiterten Portal-Module gehören verschiedenste Software-Anwendungen, die von den Portal-Herstellern als ergänzende Module für das Portal angeboten werden. Typische Beispiele hierfür sind Workflow-Management- oder Groupware-Funktionalität.

### *Prozessunterstützung*

Der Basisdienst Prozessunterstützung ist eine der jüngsten Funktionalitäten von Portal-Software. Dieser gestattet eine visuelle Modellierung des Prozesses, der im Portal abgebildet werden soll. Hierbei wird die Portal-Anwendung mit Hilfe des Modellierungswerkzeuges, den Daten der im Portal integrierten Systeme und anwendungsspezifischer Logik visuell und durch Programmierung entwickelt.

## 2.1.6. Integration von Anwendungen in Portale

Die Integration bezeichnet die Prozessabbildung und den Datenaustausch zwischen verschiedenen heterogenen Anwendungen über eine Portal-Plattform und stellt eine Grundlage für Portale dar. Aus diesem Ansatz heraus benötigen diese Integrationswerkzeuge. Die Integration von Anwendungen in ein Portal kann auf der Basis verschiedener Ansätze erfolgen. Zum einen durch die anwendungsorientierte Integration und zum anderen durch die Datenintegration. [6]

### 1.) Anwendungsorientierte Integration

Die anwendungsorientierte Integration führt Anwendungen zusammen, die in der Lage sind, HTML-basierte Ausgaben zu erzeugen. Dazu zählen alle webfähigen Anwendungen. Dies ist der einfachste Fall, da hier zur Integration die Ausgabe des Systems herangezogen werden kann und lediglich der eigentliche HTML-Inhaltsteil in die zu erstellende Portal-Seite eingefügt werden muss. Hierbei ist zu beachten, dass Design und Layout bereits auf Anwendungsseite angepasst sind oder durch das Portal anpassbar sein müssen. Bei dieser Integration entspricht die Portalanwendung direkt dem integrierten System.

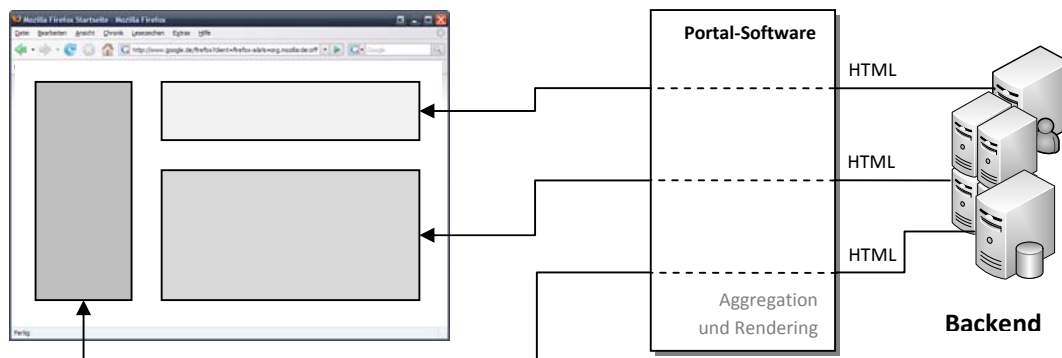


Abbildung 6: Anwendungsorientierte Portal-Integration (Quelle: [40])

### 2.) Datenintegration

Werden für das Portal spezielle Anwendungen entwickelt, die Informationen aus verschiedenen Systemen beziehen, ist eine Datenintegration notwendig. Bei diesem Ansatz werden Systeme an die Portal-Software gekoppelt, sodass Portal-Anwendungen in der Lage sind auf die verschiedenen Backend-Systeme zuzugreifen und deren Informationen darzustellen. Dafür werden die vom Application-Server bereitgestellten Integrationsdienste benutzt. Diese Integrationsform im Backend der Portale, ist für den Portal-Nutzer nicht

sichtbar, dafür umso mächtiger. Er profitiert von Portal-Anwendungen, deren Funktions- und Informationsumfang die Summe mehrere Anwendungen ist.

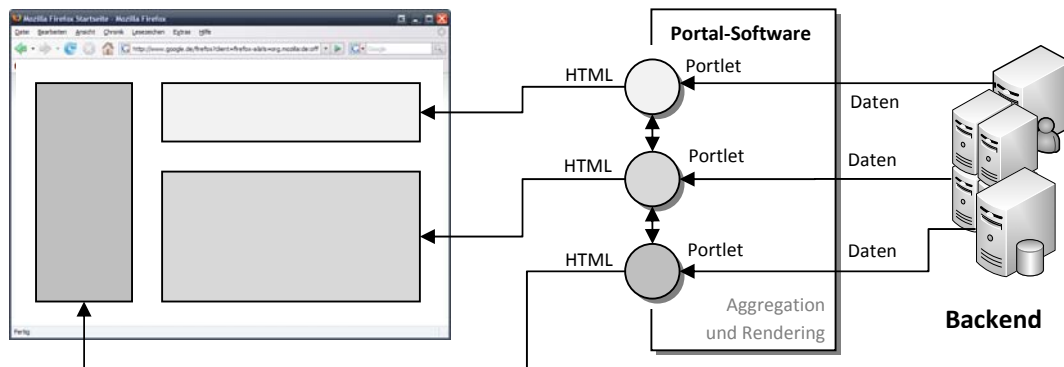


Abbildung 7: Datenintegration in das Portal (Quelle: [40])

## 2. 1. 7. Klassifizierung von Portalen

Es gibt unterschiedliche wissenschaftliche Ansätze, um Portale nach verschiedenen Kriterien zu typisieren und zu kategorisieren. Man kann Portale nach der Breite der erreichbaren Themengebiete, dem Nutzerkreis, nach der netzwerktechnischen Erreichbarkeit und dem Funktionsschwerpunkt klassifizieren. [7]

### 1.) Kategorisierung nach der Breite der mit Hilfe des Portals erreichbaren Themengebiete

#### *Horizontale Portale (Hortals)*

Horizontale Portale sind allgemeine Einstiegsseiten ins WWW, welche auf ein umfassendes Themenspektrum ausgelegt sind. Sie richten sich eher an ein Massenpublikum oder die gesamte Zielgruppe im Konsumenten-Bereich, indem sie über alle Branchen hinweg Angebote in Katalogen zusammenfassen und eventuell untereinander vergleichen. Sie sind ursprünglich als Suchmaschinen gestartet worden. Typische Bestandteile des horizontalen Portals sind aktuelle allgemeine Nachrichten, Sport, Wetter, Kino, Shopping, Auktionen, Suchmaschinen und E-Mail-Service. Der Kunde befindet sich bei horizontalen Portalen bildlich gesehen auf der Durchreise und wird an die eigentlichen Anbieter von Informationen und Dienstleistungen weitergeleitet. Der Vorteil für den Besucher von horizontalen Portalen liegt hauptsächlich darin, dass es eine deutliche Vereinfachung ist, die gewünschten Informationen in der Informationsflut des Internets zu finden. Zudem wird das Portal aufgrund der Personalisierung zur praktischen, individuellen Startseite mit der man einen Überblick über die selektierten Informationen hat. Beispiele für horizontale Portale sind *Yahoo!*, *web.de* oder *T-online.de*.

#### *Vertikale Portale (Vortals)*

Diese Art von Portalen wendet sich einer speziellen Interessengruppe zu und beschränkt sich auf einzelne Marktsegmente, Fächer, Themen, Disziplinen, Branchen, Wertschöpfungsketten und Geschäftspartner. Sie nehmen sich einen Teil aus dem horizontalen Spektrum, um ihn dann in diesem Bereich tiefgründiger, man kann auch sagen vertikal, aufzuarbeiten. Beispiele hierzu sind Portale, die Themen für Minderheiten aufgreifen oder sich direkt an einzelne Branchen richten. Im Prinzip kann man zu jedem beliebigen Thema ein vertikales Portal erstellen. Ein weiterer Vorteil dieser Portale ist,

dass die Dienste, die angeboten werden, genauer auf die Zielgruppe zugeschnitten werden können, da sie ja eine spezifische Nutzergruppe haben. Unternehmens-Portale sind klassische vertikale Portale (vgl. Kapitel 3.6.2).

## **2.) Kategorisierung nach dem Nutzerkreis**

### *Offene Portale (Web-Portale)*

Offene Portale sind grundsätzlich für jeden Benutzer über das Internet oder Intranet zugänglich. Oft dienen sie der strukturierten Recherche und Präsentation von Inhalten im WWW.

### *Geschlossene Portale*

Geschlossene Portale sind auf spezifische Bedürfnisse ausgerichtet. Sie stehen nur einer definierten Benutzergruppe im Internet oder Intranet zur Verfügung. Um Zugang zu einem geschlossenen Portal zu erlangen, müssen sich die Benutzer immer erst authentifizieren.

## **3.) Kategorisierung nach der netzwerktechnischen Erreichbarkeit**

### *Intranet-Portale*

Intranet-Portale sind in der Regel nicht öffentlich zugänglich und meist liegt der Fokus auf der Nutzung innerhalb einer Institution. Der Umfang an Funktionalität variiert von der Unterstützung bei der Arbeit bis hin zu voll integrierten Arbeitsumgebungen.

### *Extranet-Portale*

Die Extranet-Portale sind den Intranet-Portalen sehr ähnlich. Sie haben einen erweiterten Fokus auf externe Mitarbeiter und Partner.

### *Internet-Portale*

Unter Internet-Portalen sind Internet-Seiten zu verstehen, die als Plattformen für Informationen aus dem Internet allgemein oder spezifisch eingesetzt werden. Internet-Portale sind öffentlich zugänglich und unterliegen meist keinem beschränkten Personenkreis.

## 2. 1. 8. Abgrenzung des Portalbegriffs

Portale werden oft im Zusammenhang mit verschiedenen Begriffen verwendet. Deshalb wird im Folgenden versucht, Portale von einigen anderen Begriffen abzugrenzen. [8]

### 1.) Im Browser eingestellte Startseiten

Die in einem Browser eingestellte Startseite ermöglicht dem Nutzer den Zugang zu den Informationen des WWW. Obwohl diese Seite dem Benutzer beim Einstieg in das Web hilft, ist sie kein Portal. Sie dient nur diesem einen und nicht einer Vielzahl an Nutzern.

### 2.) Homepages

Eine Homepage ist eine Einstiegsseite zur Webpräsenz eines Unternehmens, einer Organisation oder einer Person. Eine Homepage ermöglicht zwar den Zugang zu den im Web zur Verfügung gestellten Ressourcen, aber als Portal kann es deshalb noch nicht bezeichnet werden. Ein Portal ist durch eine hohe Nutzungsintensität, einen großen Funktionsumfang und Personalisierungsmöglichkeiten gekennzeichnet. Erst wenn eine Homepage diese Fähigkeiten aufweist, kann sie als Portal bezeichnet werden.

### 3.) Suchmaschinen

Es gibt im Wesentlichen zwei Gründe, warum populäre Suchmaschinen, wie *Yahoo!* und *Web.de*, im WWW als Portale bezeichnet werden. Zum Einen nutzen viele Menschen die Suchmaschinen als Ausgangspunkt, um im Web nach Informationen zu suchen. Deshalb weisen diese Seiten eine enorme Nutzungsintensität aus. Zum Anderen haben die Betreiber ihre Angebote in den letzten Jahren sehr stark erweitert, sodass den Nutzern heute nicht nur Suchfunktionen, sondern ein breites Angebot an Informationen und Funktionen zur Verfügung steht. Das letzte Argument führt dazu, dass *Google* lange Zeit nicht als Portal bezeichnet werden konnte, da die Betreiber das Funktionsspektrum bewusst eng gestalteten und sich im Wesentlichen auf Suchfunktionalitäten beschränkten. Erst in der letzten Zeit wurden weitere Funktionen ergänzt, sodass auch bei *Google* eine Entwicklung von der reinen Suchmaschine hin zu einem Portal zu bemerken ist. Die von Portalen geforderte Personalisierungsmöglichkeit erfüllt *iGoogle*<sup>1</sup> im vollen Maße.

---

<sup>1</sup> Link: <http://www.google.de/ig>

## **2. 1. 9. Zusammenfassung**

Ein Portal ist der Zugangspunkt, über den ein Nutzer eines IT-Systems alle für ihn relevanten Informationen und Anwendungen angeboten bekommt, um einer bestimmten Aufgabe oder einem Wunsch nachzugehen. Portale bringen Ordnung in die Informationslandschaft, weil sie einen zentralen, einheitlichen und systemunabhängigen Zugang zu Informationen und Diensten schaffen. Sie sorgen für Orientierung in der stetig wachsenden Flut an Informationen, indem sie eine kontextabhängige Informationsrecherche anbieten und die Informationen nach ihrer Relevanz filtern, sortieren und gruppieren. Sie lenken den Blick des Nutzers weg von der Technik der Systeme und hin zu der Nutzen stiftenden Fachlichkeit.



## 2.2. Web-Technologien

Dieses Kapitel betrachtet die für Portale grundlegenden Technologien.

### 2.2.1. Basistechnologien

Das World Wide Web ist ein System zur Darstellung von Dokumenten im HTML-Format, mit der Möglichkeit diese über das Internet zu verknüpfen. Für die Realisierung des WWW werden folgende, grundlegende Komponenten benötigt. [9] [10]

#### 1.) *Web-Browser*

Eine Software für die Darstellung von Web-Inhalten in Form von Web-Seiten auf dem Bildschirm des Rechners eines Nutzers ist eine grundsätzliche Voraussetzung. Diese Software bezeichnet man als Web-Browser. Ein Browser zeigt die angeforderte Web-Seite an und bietet zudem zahlreiche Funktionen für die Navigation im Web.

#### 2.) *Web-Adresse*

Weiterhin benötigt man eine eindeutige Web-Adresse zur Angabe der Lokation von Web-Inhalten, die auch Web-Ressourcen genannt werden. Als einheitliche Web-Adressen werden sogenannte Uniform-Resource-Locator (URL)<sup>1</sup> verwendet.

#### 3.) *Web-Server*

Unabdingbar ist ein Web-Server, über den auf Web-Inhalte im Internet zugegriffen werden kann. Ein Web-Server liefert zwar vorwiegend statische, doch zunehmend auch dynamische Daten. Dynamische Daten werden erst beim Aufruf erzeugt. Dies geschieht durch den Einsatz von server-seitigen Scripten (vgl. Kapitel 2.2.2).

#### 4.) *Seitenbeschreibungssprache*

Eine weitere erforderliche Grundkomponente des WWW ist eine abstrakte Sprache für die Beschreibung von Web-Seiten. Eine Web-Seite besteht in der Regel aus mehreren Web-Objekten und wird als Hypertext dargestellt. Für die Darstellung von Web-Seiten

---

<sup>1</sup> RFC: <http://tools.ietf.org/html/rfc3986>

verwendet man die Hypertext-Markup-Language (HTML)<sup>1</sup>. HTML ist eine Auszeichnungssprache und legt fest, wie die Informationen gegliedert und die Dokumente verknüpft sind.

### 5.) Transportprotokoll

Ebenfalls unentbehrlich ist ein Protokoll für den Transport von Web-Inhalten zwischen Browsern und Web-Servern. Dafür wird das Hypertext-Transfer-Protocol (HTTP)<sup>2</sup> verwendet. Wenn ein Nutzer eine Web-Seite angefordert hat, sendet sein Browser einen HTTP-Request an den durch die URL angegebenen Web-Server. Er empfängt diese Anforderung und sendet eine HTTP-Response, in der sich der angeforderte Web-Inhalt befindet, an den Browser zurück.

Für die Übertragung der Web-Inhalte zwischen Web-Server und Web-Browser nutzt HTTP das verbindungsorientierte Transmission-Control-Protocol (TCP)<sup>3</sup>.

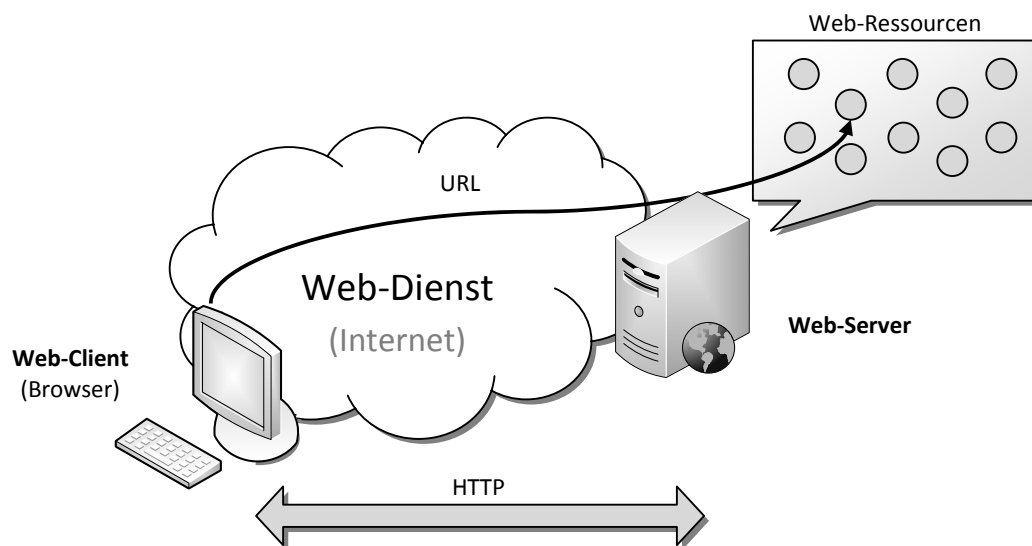


Abbildung 8: Grundkomponenten des Web

<sup>1</sup> W3C Recommendation: <http://www.w3.org/TR/html4/>

<sup>2</sup> RFC: <http://tools.ietf.org/html/rfc2616>

<sup>3</sup> RFC: <http://tools.ietf.org/html/rfc793>

## 2.2.2. Scriptsprachen

Scriptsprachen sind Programmiersprachen, die für kleine, überschaubare Programmieraufgaben gedacht waren. Sie verzichten häufig auf bestimmte Sprachelemente, deren Nutzen erst bei der Bearbeitung größerer Projekte zum Tragen kommt. Man unterscheidet zwischen *server-* und *client-seitigen Scriptsprachen*. [10]

### 1.) Server-seitige Scriptsprachen

Server-seitige Scriptsprachen werden dazu genutzt, um dynamisch HTML-Seiten zu generieren. Dazu verwendet man meist in HTML eingebettete HTML-generierenden Scripte. Diese Scripte werden entweder interpretiert oder kompiliert. Mit *SSI*<sup>1</sup>, *ASP*<sup>2</sup> und *PHP*<sup>3</sup> geschriebene Scripte werden interpretiert, demgegenüber Scripte in *JSP*<sup>4</sup> kompiliert werden.

### 2.) Client-seitige Scriptsprachen

Client-seitige Scriptsprachen werden nicht auf dem Server, sondern lokal in den Browsern ausgeführt. Die bekannteste client-seitige Scriptsprache ist *JavaScript*. Im Gegensatz zu den server-seitigen, dienen client-seitige Scripte zwar nicht ausschließlich, aber hauptsächlich dazu, HTML-Seiten zu ergänzen.

---

<sup>1</sup> Server-Side-Includes; Link: <http://httpd.apache.org/docs/howto/ssi.html>

<sup>2</sup> Active-Server-Pages; Link: <http://www.asp.net/>

<sup>3</sup> PHP-Hypertext-Preprocessor; Link: <http://de.php.net/>

<sup>4</sup> JavaServer-Pages; Link: <http://java.sun.com/products/jsp/index.html>

## 2.2.3. XML

Die Extensible-Markup-Language (XML)<sup>1</sup> definiert eine Syntax, um strukturierte Datenbestände jeder Art mit einfachen, verständlichen Auszeichnungen zu versehen, die zugleich von Anwendungen der unterschiedlichsten Art ausgewertet werden können. Dieser Syntax ist einfach und zugleich streng. Der Anwender kann sich so seine eigenen Vokabulare zur Beschreibung aufbauen oder auf existierende Vokabulare zurückgreifen. [11]

### 1.) Struktur

XML-Dokumente besitzen einen physischen und einen logischen Aufbau. Der physische Aufbau eines XML-Dokumentes besteht aus Entitäten, einer optionalen XML-Deklaration und einer optionalen Dokumenttyp- beziehungsweise XML-Schema-Definition.

Der logische Aufbau eines XML-Dokumentes ähnelt einer Baumstruktur und ist damit hierarchisch strukturiert. Als Baumknoten können Elemente, Attribute, Verarbeitungsanweisungen, Kommentare und Text auftreten. Ein XML-Dokument muss genau ein Element auf der obersten Ebene enthalten. Unterhalb von diesem Dokument-Element können weitere Elemente verschachtelt werden.

### 2.) Schemasprachen

Um die Struktur von XML-Sprachen zu beschreiben, nutzt man so genannte Schemasprachen. Die zwei bekanntesten sind die *Dokumenttyp-Definition* (DTD) und die *XML-Schema-Definition* (XSD). [11]

Eine Dokumenttyp-Definition legt die zu verwendenden Tags fest und bestimmt, in welcher Reihenfolge sie auftreten und wie sie verschachtelt werden dürfen. In einer DTD ist es nicht möglich, zwischen Texten und Zahlen zu unterscheiden. Ein weiterer Nachteil ist die Tatsache, dass die DTD in einer eigenen Sprache abgefasst werden muss. Zudem kennt die DTD keine Namensräume.

Eine XML-Schema-Definition ist die moderne Möglichkeit, die Struktur von XML-Dokumenten zu beschreiben und bietet zudem die Möglichkeit, den Inhalt von Elementen

---

<sup>1</sup> W3C Recommendation: <http://www.w3.org/XML/>

und Attributen mittels regulärer Ausdrücke zu beschränken, etwa auf Zahlen, Datumsangaben oder Texte. Eine XSD ist selbst ein XML-Dokument, welches erlaubt, komplexere und inhaltliche Zusammenhänge zu beschreiben als dies mit einer DTD möglich ist.

### 3.) XSL und XSLT

Die Extensible-Stylesheet-Language (XSL)<sup>1</sup> wurde als allgemeine Stylesheet-Sprache für XML-Daten geschaffen. Die grundsätzliche Aufgabe von XSL ist es, XML-Daten zu formatieren. Dazu stellt XSL ähnlich wie CSS<sup>2</sup> einen großen Umfang an Style-Eigenschaften bereit und enthält darüber hinaus aber auch Möglichkeiten, die den logischen Ablauf der Datenpräsentation steuern. XSL kann zwar noch weitgehend als Beschreibungssprache für Formate bezeichnet werden, enthält darüber hinaus aber auch Konstrukte wie bedingungsabhängige Anweisungen oder Schleifenanweisungen, die eher an Programmiersprachen erinnern.

Mit der Transformations-Komponente XSLT<sup>3</sup> verfügt XSL aber noch über eine weitere wichtige Funktion. XSLT erlaubt es, ein XML-Dokument in ein anderes zu transformieren. Die vorzunehmenden Veränderungen werden in einem wohlgeformten XML-Dokument in Form von Templates, sogenannten XSLT-Stylesheets, zusammengestellt. Die nachfolgende Abbildung illustriert dies.

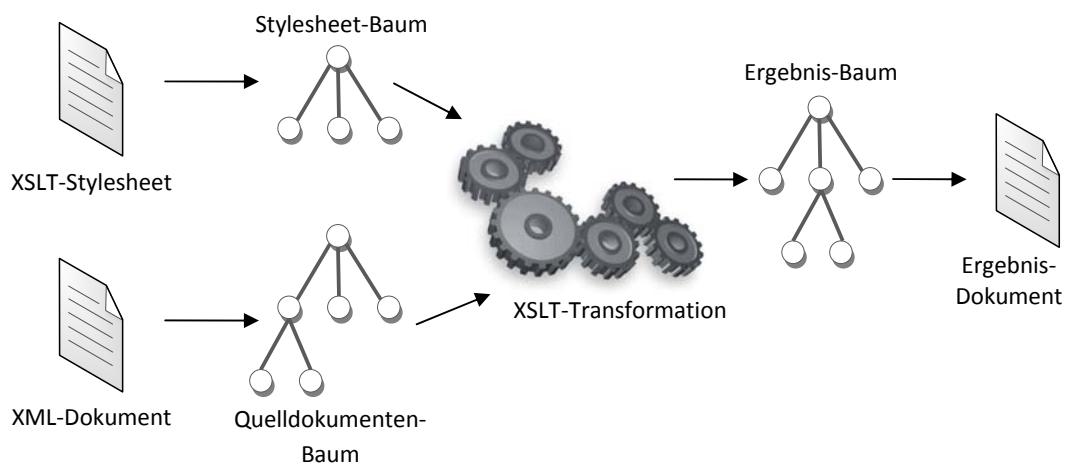


Abbildung 9: Schema der Transformation (Quelle: [43])

<sup>1</sup> W3C Recommendation: <http://www.w3.org/Style/XSL/>

<sup>2</sup> Cascading-Style-Sheets, deklarative Stylesheet-Sprache, W3C Recommendation: <http://www.w3.org/Style/CSS/>

<sup>3</sup> XSL-Transformations, W3C Recommendation: <http://www.w3.org/TR/xslt>

#### 4.) Programmierschnittstellen

Als Schnittstelle für den Zugriff auf XML-Dokumente hat das W3C eine Spezifikation für das *Document-Object-Model* (DOM)<sup>1</sup> verabschiedet. Die DOM-Schnittstelle repräsentiert das XML-Dokument als eine Baumstruktur und erlaubt es gezielt auf Teile des Dokuments zuzugreifen.

*Simple-API-for-XML* (SAX)<sup>2</sup> ist eine weitere Programmierschnittstelle für XML-Dokumente. Die SAX-API repräsentiert ein XML-Dokument als sequentiellen Datenstrom und ruft für definierte Ereignisse vorgegebene Rückruf-Funktionen auf. Solche Ereignisse können etwa das Erreichen des Anfangs und des Endes jedes Elementes sein.

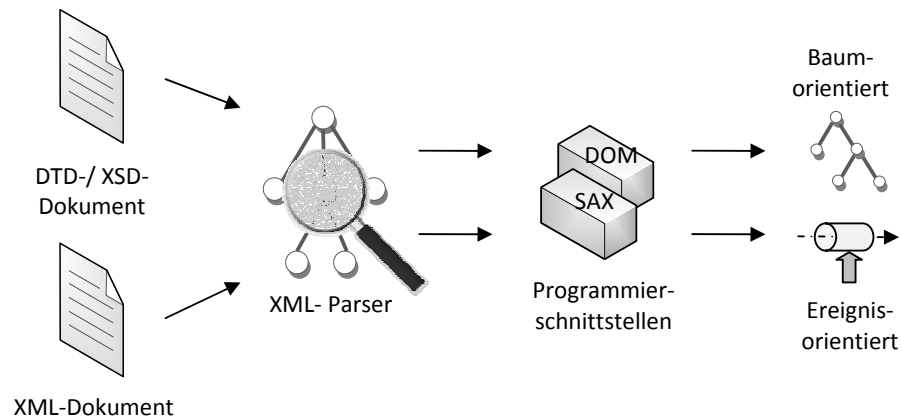


Abbildung 10: XML - Programmierschnittstellen

Nachfolgende Tabelle gibt einen Überblick der Vor- und Nachteile beider Programmiermodelle.

<sup>1</sup> W3C Recommendation: <http://www.w3.org/DOM/>

<sup>2</sup> Link: <http://www.saxproject.org/>

	<b>DOM</b>	<b>SAX</b>
<b>Vorteile</b>	<ul style="list-style-type: none"> <li>• Implementierung eines mächtigen Standards</li> <li>• Lese- und Schreibzugriffe auf beliebige XML-Elemente möglich</li> </ul>	<ul style="list-style-type: none"> <li>• Geringerer Speicherverbrauch, da immer nur das aktuelle XML-Element verfügbar ist</li> <li>• schnell</li> </ul>
<b>Nachteile</b>	<ul style="list-style-type: none"> <li>• Langsam und speicherintensiv, da vollständiger Baum von Objekten erzeugt werden muss</li> </ul>	<ul style="list-style-type: none"> <li>• Lineare Verarbeitung, kein beliebiger Zugriff auf Elemente möglich</li> <li>• Kein Schreibzugriff möglich</li> </ul>

Tabelle 1: Vor- und Nachteile der Programmiermodelle DOM und SAX

### 5.) Zusammenfassung

XML bietet eine einheitliche Methode für die Beschreibung und den Austausch strukturierter Daten, unabhängig von einer bestimmten Plattform oder Programmiersprache. Dabei wird der Inhalt völlig von der Form der Darstellung getrennt, die aber jederzeit über Stylesheets zugeordnet werden kann.

## 2.2.4. Web 2.0

Das Web 2.0 ist geprägt durch ein neues Netzverständnis bestehend aus einer veränderten Wahrnehmung und Benutzung des Webs, mit dem Hauptaspekt aus organisatorischer Sicht. Inhalte werden nicht mehr nur zentralisiert von großen Medien erstellt und an die Massen verbreitet, sondern auch von unabhängigen Leuten, die sich untereinander vernetzen. Typische Beispiele hierfür sind *Wikis*, *Weblogs* sowie *Bild-* und *Videoportale*<sup>1</sup> und *Tauschbörsen*.

### 1.) Definition

Tim O'Reilly veröffentlichte im Jahre 2005 einen richtungsweisenden Artikel mit den Titel „*What is Web 2.0*“ [12] und legte dar, was grundlegend unter dem Begriff Web 2.0 zu verstehen ist. Dazu verglich er einige Anwendungen des Web 1.0 mit denen des Web 2.0 und arbeitete sieben Schlüsselprinzipien des Web 2.0 heraus.

#### *The Web as Platform*

Das Internet ist mittlerweile allgegenwärtig. Jeder kennt es und jeder nutzt es und das nicht nur über den Browser sondern über viele Dienste und Praktiken.

#### *Harnessing Collective Intelligence*

Kollektive Intelligenz wird durch die Vernetzung der Daten und Seiten im Web genutzt.

#### *Data is the next Intel Inside*

Die durch eine Web-Anwendung gesammelten Daten sind die Basis der Web-Anwendung selbst. Eine große Menge von Daten wird durch eine Anwendung der Masse preisgegeben.

#### *End of the Software Release Cycle*

Software wird nicht mehr als Produkt ausgeliefert sondern als Service. Ein Produkt aktuell zu halten ist wesentlich schwieriger als einen Service.

#### *Lightweight Programming Models*

---

<sup>1</sup> Bildportal: Flickr r, Link: <http://www.flickr.com/>; Videoportal: YouTube, Link: <http://www.youtube.com/>



Um Daten und Services der breiten Menge bereit zu stellen werden leichtgewichtige Programmiermodelle implementiert. Das heißt, dass die Daten sehr einfach über eine HTTP- oder Web-Service-Schnittstelle bereitgestellt werden.

### *Software Above the Level of Single Device*

Nicht nur der PC ist als Endgerät geeignet, sondern auch mobile oder sonstige Geräte.

### *Rich User Experience*

Das Web ist nicht mehr nur einfaches HTML mit einer einfachen Informationsseite sondern, es entstehen Anwendungen die normalen Desktop-Anwendungen um nichts nachstehen.

Der Artikel beschreibt diese sieben Prinzipien sehr ausführlich und mit einigen Beispielen. Allerdings ist es dennoch recht schwierig den Begriff Web 2.0 zu verstehen und aus dem Grund verfasst O'Reilly in seinem Blog eine kompakte Definition, die das Web 2.0 gut beschreibt.

*„Web 2.0 is the network as platform, spanning all connected devices; Web 2.0 applications are those that make the most of the intrinsic advantages of that platform: delivering software as a continually-updated service that gets better the more people use it, consuming and remixing data from multiple sources, including individual users, while providing their own data and services in a form that allows remixing by others, creating network effects through an “architecture of participation,” and going beyond the page metaphor of Web 1.0 to deliver rich user experiences.” [13]*

In dieser Definition sind noch mal alle wichtigen Prinzipien zusammengefasst. Bei dem Web 2.0 handelt es sich um ein Netzwerk als Plattform für Applikationen, welche als Dienste veröffentlicht werden und ständigen Updates unterliegen. Wichtig dabei ist die Beteiligung der Nutzer, denn sie beeinflussen die Inhalte selbst. Durch die Vernetzung und Mitwirkung aller Beteiligten werden Dienste zu so genannten *Mashups*<sup>1</sup> kombiniert. Synergie-Effekte führen dabei zu einem Resultat, das mehr als die Summe der verwendeten Komponenten ist und weit über den Ausdruck der Web-Seiten des Netzes der ersten Generation hinausgeht und ein größeres Erlebnis für den Nutzer bietet. [14]

---

<sup>1</sup> Ein Mashup ist eine Seite die sich aus mehreren schon vorhandenen Teilen anderer Web-Seiten zusammensetzt.

## **2.) Web 2.0 Anwendungen**

Nachfolgend werden Anforderungen an Anwendungen definiert, damit diese dem Web 2.0 zugeordnet werden können. [15]

### *Internetfähigkeit*

Eine Web 2.0 Anwendung ist eine Internet-Anwendung, die über einen Browser genutzt werden kann.

### *Dynamisches Verhalten*

Die Web-Seiten einer Web 2.0 Anwendung werden dynamisch erzeugt, nicht statisch geladen. Sie verändern sich beispielsweise je nach Benutzereingabe.

### *Ausdrucksmöglichkeit*

Den Benutzern wird die Möglichkeit gegeben, selbst Beiträge einzustellen um ihr Wissen und ihre Ideen einzubringen.

### *Rechtesystem*

Mit einem fein ausgearbeiteten Rollen- und Rechtesystem kann unterschieden werden, wer welche Beiträge welcher Personen einsehen oder bearbeiten kann. Nutzer können dafür in Gruppen mit verschiedenen Rechten eingeteilt werden.

### *Persönliche Profile*

Jeder Nutzer bekommt das Recht, bestimmte vorgefertigte persönliche Datensätze auszufüllen. Dies wird oft als persönliches Profil bezeichnet.

### *Personalisierung*

Die Web-Seiten einer Web 2.0 Anwendung können in einem sehr geringem Umfang und nahezu nur oberflächlich an die eigenen Vorlieben angepasst werden.

### *Erweiterbarkeit*

Jeder Beitrag kann von jedem um eine Anmerkung erweitert werden.

### **3.) Portale als Web 2.0 Anwendungen**

Betrachtet man die sieben Schlüsselprinzipien [12] und die Anforderungen an Anwendungen des Web 2.0 [15], so kommt man zu dem Schluss, dass Portale Web 2.0 Anwendungen sind.

Tim O'Reilly definierte als erstes Prinzip das Internet als Plattform des Web 2.0. Portale sind meistens im Web angesiedelt, auch wenn dies nicht zwingend erforderlich ist. Weiterhin sind Portale der Zugangspunkt zu Informationen und Diensten. Das zweite Prinzip der Nutzung von kollektiver Intelligenz ist damit auch erfüllt. Die durch ein Portal gesammelten Daten machen den Wert eines Portals aus. Ein Portal lebt von seinen Nutzern beziehungsweise deren preisgegebenen Wissen. Im Fall von Web 2.0 spricht man auch von datengetriebenen Anwendungen. Ein Portal ist zudem kein fertiges Produkt, sondern eher ein Portfolio an Diensten. Weiterhin verwenden Portale leichtgewichtige Programmiermodelle, indem sie verteilte Daten und Dienste einbinden. Zwei weitere Prinzipien des Web 2.0 sind damit erfüllt. Portale ermöglichen die Nutzung über die Grenzen einzelner Geräte hinaus. Möglich macht dies die Präsentationsschicht und deren visuelle Darstellung des Portals für verschiedene Frontends. Zuletzt wird noch ein umfassendes Bedienkonzept einer Web 2.0 Anwendung verlangt, was dem einer Desktop-Anwendung gleicht. Die Entwicklung von Webtechnologien geht stetig voran, sodass Web-Anwendungen bald nicht mehr von Desktop-Anwendungen zu unterscheiden sind. Portale erfüllen somit auch die letzten zwei Schlüsselprinzipien.

Betrachtet man anschließend die sieben Anforderungen an eine Web 2.0 Anwendung, so wird die Aussage bestätigt. Ein Portal ist internetfähig, dynamisch, bietet gewisse Ausdrucksmöglichkeiten für den Nutzer, ein Rechtssystem, persönliche Profile und die Möglichkeit der Personalisierung und Erweiterbarkeit.

### **4.) Zusammenfassung**

Der Unterschied von Web 2.0 Anwendungen zu herkömmlichen Web-Anwendungen ist, dass jeder Nutzer sein Wissen und seine Ideen in eine bestimmte Web-Anwendung einbringt und mit anderen Nutzern teilt. Die Folge ist ein Erlebnis für den Nutzer, das weit über den Ausdruck der Web-Seiten der ersten Generation hinausgeht.

## 2.2.5. SOA

Bei der Service-orientierten Architektur (SOA) handelt es sich nicht um eine konkrete Technik, sondern um eine Abstraktion, die im gegebenen Zusammenhang wesentliche Aspekte hervorhebt und unwesentliche unterdrückt.

### 1.) Grundlegendende Merkmale einer SOA

Nachfolgend werden die 5 wichtigsten Aspekte einer Service-orientierten Architektur vorgestellt. [16]

Das erste Merkmal einer SOA ist die *lose Kopplung* der Dienste. Dienste werden bei Bedarf dynamisch gesucht, gefunden und eingebunden.

Damit Dienste dynamisch eingebunden werden können, ist ein *Verzeichnisdienst* oder *Repository* notwendig. Dort werden die zu Verfügung gestellten Dienste registriert. Zudem ermöglicht es die Suche nach Diensten.

Möchte eine Applikation einen Dienst nach erfolgreicher Suche nutzen, so muss die Applikation in der Lage sein, sich mit dem Dienst zu unterhalten. Notwendige Voraussetzung dafür sind *offene Standards*, damit der Nutzer den Dienst eines unbekannten Anbieters auch verstehen kann.

Die *Einfachheit* ist ein weiteres wichtiges Merkmal einer SOA. Die Nutzung gekapselter Dienste gestattet es, diese in verschiedenen Umgebungen mehrfach ohne Aufwand wiederzuverwenden.

Die Einfachheit der SOA führt dazu, eine Umsetzung schnell voran zu treiben und die Akzeptanz zu fördern. Trotzdem sollte man das Thema *Sicherheit* nicht außer Acht lassen. Sicherheitsbelange wie Vertraulichkeit und Berechtigung müssen berücksichtigt werden.

Die nachfolgende Abbildung zeigt das Grundkonzept einer SOA. Das Fundament wird von offenen Standards, Sicherheit und Zuverlässigkeit gebildet. Die verteilten Dienste, die lose Kopplung, die Plattformunabhängigkeit und die prozessorientierte Struktur sind die tragenden Säulen.

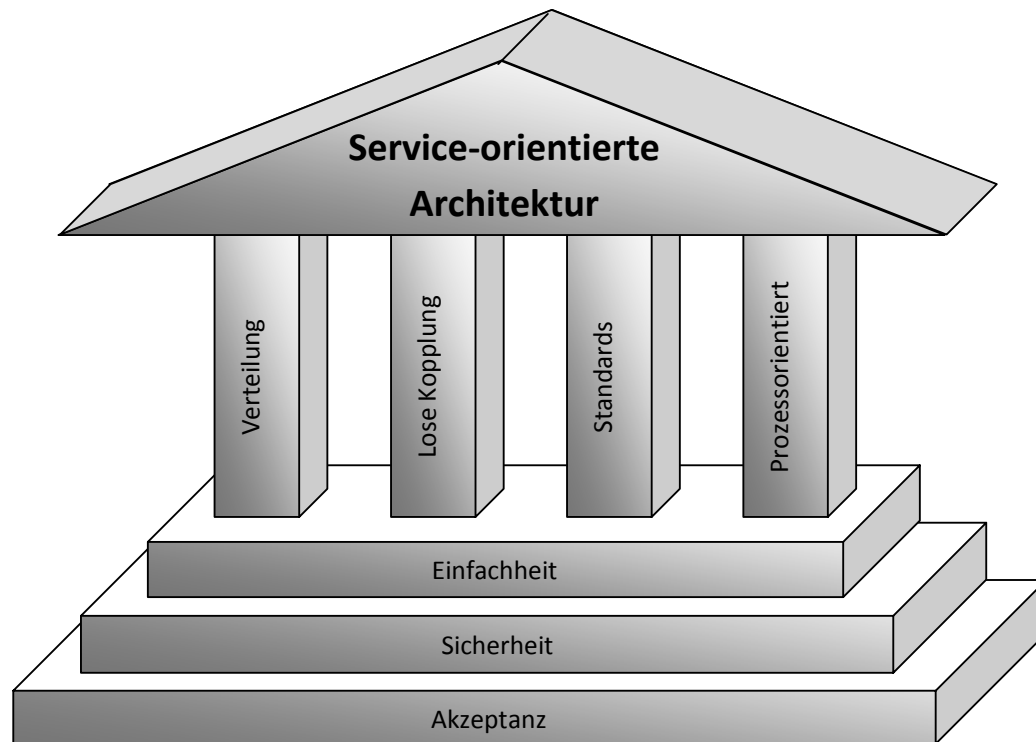


Abbildung 11: SOA- Tempel (Quelle: [47])

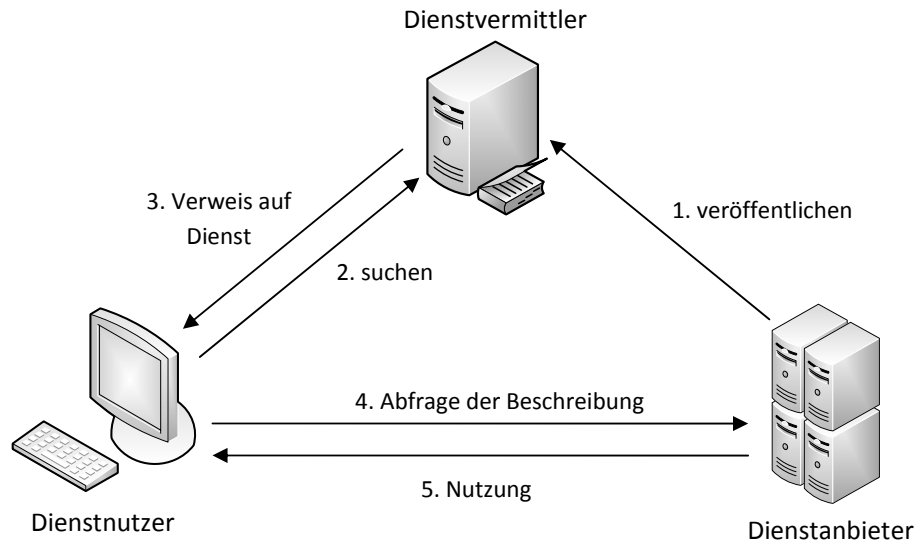
## 2.) Definition

Aus den oben beschriebenen Merkmalen kann folgende Definition einer Service-orientierten Architektur abgeleitet werden.

*„Unter einer SOA versteht man eine Systemarchitektur, die vielfältig, verschiedene und eventuell inkompatible Methoden oder Applikationen als wiederverwendbare und offen zugreifbare Dienste repräsentiert und dadurch eine plattform- und sprachenunabhängige Nutzung und Wiederverwendung ermöglicht“ [16]*

## 3.) Rollenmodell

Die SOA definiert drei Rollen bei der Anwendung von Diensten.



**Abbildung 12: Rollenmodell der SOA (Quelle: [16])**

Diese Rollen sind zum einen der Dienstanbieter und zum anderen den Dienstvermittler, bei dem der Dienstanbieter seinen Dienst registriert. Die dritte Rolle nimmt der Dienstnutzer ein, der beim Dienstvermittler einen Dienst suchen kann, woraufhin er einen entsprechenden Verweis zurück bekommt. Im Anschluss daran, kann der Dienstnutzer über den Verweis beim Dienstanbieter die Dienstbeschreibung abfragen und anschließend den Dienst nutzen. Das Rollenmodell wird im nächsten Kapitel über Web Services, noch einmal genauer betrachtet.

### **4.) Zusammenfassung**

Service-orientierte Architekturen sind das abstrakte Konzept einer Softwarearchitektur, in deren Zentrum das Anbieten, Suchen und Nutzen von Diensten über ein Netzwerk steht. Wesentlicher Vorteil einer solchen Architektur ist die Unabhängigkeit von den Details der jeweiligen Implementierung. Dadurch ist eine funktionale Zerlegung der Anwendung möglich und eine prozessorientierte Betrachtungsweise wird erleichtert.

Zurzeit sind Web Services die einzige fortgeschrittene Technik zur Implementierung einer SOA. Das nächste Kapitel wird sich auf diese Technik konzentrieren und zeigen, wie eine SOA auf Basis von Web Services ermöglicht werden kann.

## 2.2.6. Web Services

In dem vorherigen Kapitel wurde der Begriff der Service-orientierte Architekturen vorgestellt und erläutert. Web Services sind eine konkrete Implementierung dieses Konzeptes.

### 1.) Definition

Für eine offizielle Definition von Web Services sei auf das World-Wide-Web-Consortium (W3C) verwiesen, dessen letzte Version einer Definition vom Februar 2004 lautet:

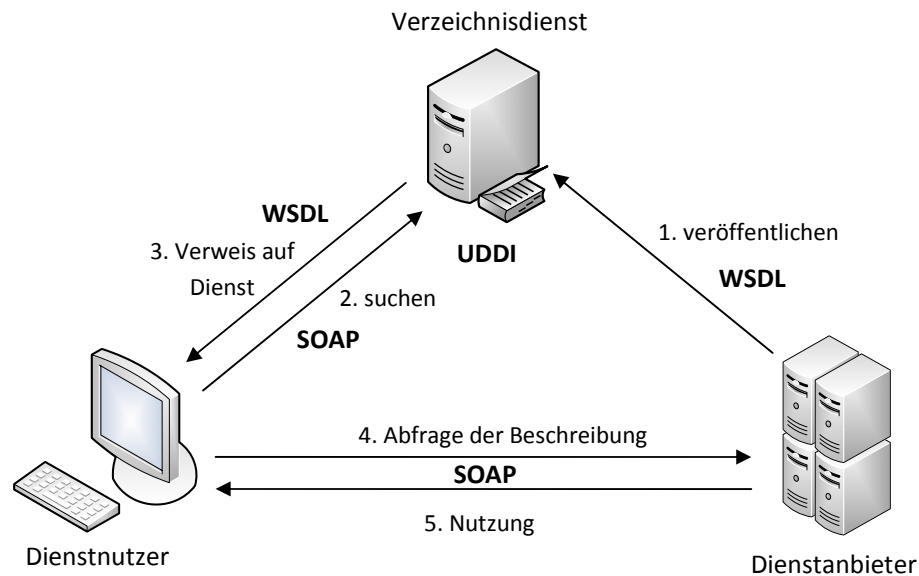
*„A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.“ [17]*

Vereinfacht lassen sich Web Services folgendermaßen definieren:

Web Services sind autonome, sich selbst beschreibende, modulare Anwendungen, die einerseits innerhalb eines großen Systems als effiziente Methode zur verteilten Verarbeitung angewendet, andererseits im Internet publiziert, lokalisiert und aufgerufen werden können.

### 2.) Web Service-Rollenmodell

Alle wesentlichen Nutzungsformen von Web Services lassen sich durch eine Reihe von Standard-Szenarien abdecken. Diese lassen sich alle in das Web-Service-Rollenmodell einpassen.



**Abbildung 13: Web-Service-Rollenmodell (Quelle: [16])**

In diesem Modell gibt es drei zentrale Rollen. Die erste Rolle übernimmt der Dienstanbieter. Er bietet einen Dienst nach außen an, der durch ein Web-Service-Description (WSD) Dokument beschrieben wird. Die zweite Rolle füllt der Dienstnutzer aus, der den Dienst nutzen möchte, um eine gewisse Anwendungsfunktionalität zu realisieren. Um den Dienst des Dienstanbieters nutzen zu können, muss dem Nutzer die WSD bekannt sein, denn nur dann kann er eine Software erstellen, die korrekt auf den Dienst zugreifen kann. Es gibt mehrere Möglichkeiten an diese WSD heranzukommen. Die Einfachste ist die direkte Kommunikation mit dem Anbieter des Dienstes, was aber nicht immer möglich ist. Daher gibt es eine dritte Rolle, den Dienstvermittler beziehungsweise der Verzeichnisdienst. Der Dienstanbieter kann einen Dienst bekannt machen, indem er das WSD-Dokument beim Dienstvermittler registriert. Über eine öffentliche Schnittstelle kann nur der Nutzer beim Vermittler nach verfügbaren Diensten suchen. Findet er einen, der seinen Anforderungen entspricht, so bekommt er die WSD und erstellt die Client-Software, die dann mit dem Dienst des Dienstanbieters interagiert.

Aus den vielen Möglichkeiten der Ausgestaltung dieses Modells ergeben sich folgende Szenarien: [18]

1. *Dienstnutzer und Dienstanbieter kennen sich vor der Interaktion*
2. *Dienstnutzer und Dienstanbieter kennen sich vorher, die WSD wird aber dynamisch geladen*



3. Es gibt mehrere Dienstanbieter, die Auswahl des Dienstes erfolgt aber manuell
4. Es gibt mehrere Dienstanbieter, die Auswahl des Dienstes erfolgt aber manuell und die Web Service Beschreibung kommt vom Dienstanbieter
5. Es gibt mehrere Dienstanbieter, die Auswahl erfolgt automatisch

Der erste Punkt ist eines der Szenarien, das im B2B<sup>1</sup> realistisch ist. Zwei Partner, die schon seit einiger Zeit zusammenarbeiten, möchten nun ihre Zusammenarbeit mittels Web Services automatisieren. Die Abbildung 14 zeigt das Vorgehen, das hierzu notwendig ist.

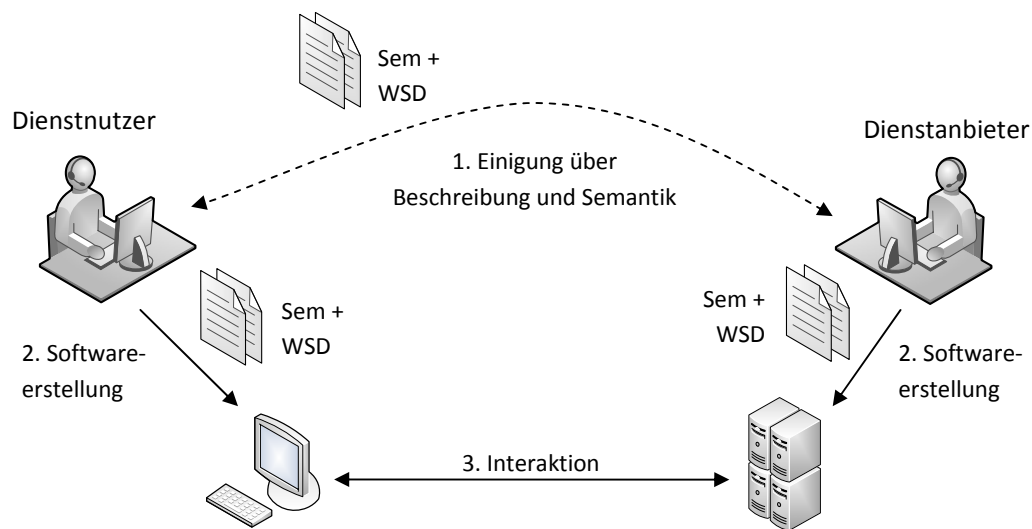


Abbildung 14: Web-Service-Szenario (Quelle: [19])

Im ersten Schritt müssen sich beide Partner über Syntax und Semantik ihrer Interaktionen verständigen. Die Syntax wird in Form eines WSD-Dokumentes und übermittelt. Da der Dienstnutzer durch eine WSD-Beschreibung nur weiß, wie er den Dienst nutzt und nicht was er macht, muss die Beschreibung noch zusätzlich semantische Information enthalten. Im zweiten Schritt müssen nun Dienstanbieter und Nutzer ihre jeweilige Software implementieren. Dazu verwenden sie dasselbe WSD-Dokument und dieselbe Information über die Semantik. Lediglich die unterschiedlichen Rollen müssen auf den jeweiligen Seiten berücksichtigt werden. Eine Interaktion der Partner kann nun beginnen.

<sup>1</sup> Business-to-Business, Bezeichnung für die computergestützte Kommunikation zwischen Unternehmen

### 3.) Web Service Basistechnologien

#### SOAP

SOAP<sup>1</sup> ist ein Protokoll für die Übertragung von Nachrichten. SOAP stützt sich auf XML zur Repräsentation der Daten und auf die Internet-Protokolle der Transport- und Anwendungsschicht zur Übertragung der Nachrichten.

Unter einer SOAP-Message wird ein SOAP-Request oder ein SOAP-Response verstanden. Wie die Abbildung 15 zeigt, besteht eine einfache SOAP-Message aus einem *SOAP-Envelope*, einem optionalen *SOAP-Header* und einem *SOAP-Body*.

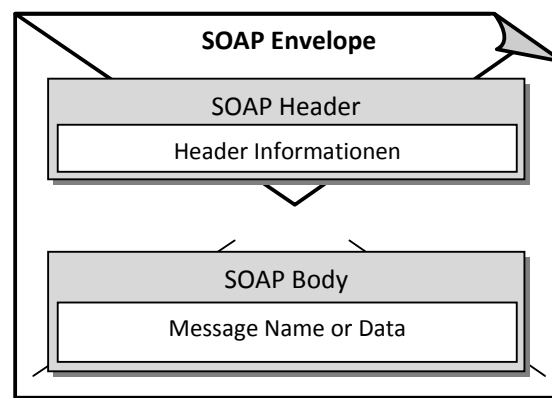


Abbildung 15: Grundlegender Aufbau einer SOAP-Message (Quelle: [16])

Der SOAP-Envelope ist eine Art Briefumschlag für SOAP-Messages und stellt das Wurzelement dar. Er legt die grundlegende Struktur einer SOAP-Message fest. Der SOAP-Body enthält die eigentliche Nachricht. Im Falle eines SOAP-Requests beschreibt die Nachricht den gewünschten Service, die aufzurufende Methode sowie die übergebenen Parameter. Der optionale SOAP-Header ermöglicht die Spezifikation zusätzlicher Informationen für die Auswertung auf Applikationsebene.

Der Transport einer SOAP-Message kann durch HTTP, SMTP, FTP oder andere Protokolle erfolgen. In der Praxis werden SOAP-Messages zumeist mittels HTTP transportiert. Ein SOAP-Request wird als Inhalt eines HTTP-Request übertragen und der SOAP-Response ist im HTTP-Response enthalten. [19]

---

<sup>1</sup> Ursprünglich: Simple-Object-Access-Protocol; Akronym wird aber nicht mehr verwendet

## WSDL

Die Web-Service-Description-Language (WSDL) spezifiziert ein XML-Format zur Beschreibung und Lokalisierung eines Web Services. Eine solche Beschreibung stellt einen Kontrakt zwischen Dienstanbieter und Dienstanutzer dar. Sie umfasst die öffentlich verfügbaren Methoden und deren Signaturen, Informationen über die unterstützten Transport-Protokolle, die Adresse des Web Services und die Typ-Schemata für den Datenaustausch. Im Grundsatz stellt WSDL eine Schnittstellenbeschreibung dar, die sich wie folgt zusammensetzt.

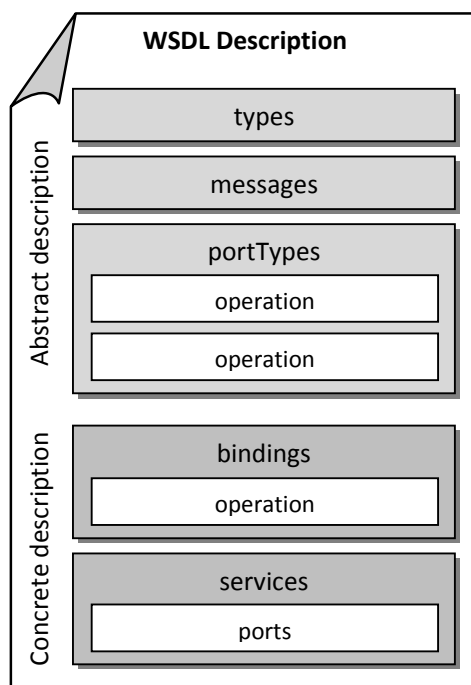


Abbildung 16: Aufbau einer WSDL-Beschreibung (Quelle: [20])

Eine WSDL-Beschreibung ist sehr modular aufgebaut. Die ersten drei Elemente stellen dabei abstrakte Beschreibungen dar. Die letzten beiden Elemente sind konkret und referenzieren zuvor beschriebene Abstraktionen.

Für die Übertragung komplexer zusammengesetzter Daten als Argumente oder Rückgabewerte, können mit dem Element `types` eigene Typen definiert werden.

Ein `message`-Element beschreibt eine Nachricht die vom Client zum Web Service oder umgekehrt übertragen werden soll. Neben dem Attribut `name` zur eindeutigen Benennung

der Nachricht kann ein `message`-Element eines oder mehrere `part`-Elemente enthalten. Jedes `part`-Element beschreibt ein Argument oder einen Rückgabewert. Neben dem Bezeichner wird auch der Typ festgelegt.

Das Element `portType` beschreibt eine abstrakte Web-Service-Schnittstelle. Eine solche Schnittstelle besteht aus einer oder mehreren abstrakten Operationen. Jede Operation setzt sich wiederum aus einer oder zwei Nachrichten zusammen. WSDL unterscheidet vier Typen von Operationen.

<b>Operation</b>	<b>Beschreibung</b>
One-way	Der Client sendet eine Nachricht an den Web Service, ohne eine Antwort zu erwarten
Request-Response	Der Client sendet eine Nachricht an den Web Service und dieser antwortet mit einer Nachricht
Solicit-Response	Der Web Service sendet eine Nachricht an den Client und bekommt von diesem eine Antwort
Notification	Der Web Service sendet eine Nachricht an den Client

**Tabelle 2: WSDL - Operationen**

Jede Operation wird durch ein `operation`-Element beschrieben. Abhängig vom Typ besteht die Operation aus einem `input`-Element und/oder aus einem `output`-Element. Eine One-way-Operation ist durch ein `input`-Element gekennzeichnet, während eine Request-Response-Operation durch ein `input`-Element gefolgt von einem `output`-Element beschrieben wird. Die Reihenfolge der beiden Elemente ist wichtig. Ein `output`-Element gefolgt von einem `input`-Element beschreibt einen Solicit-Response. Ein einzelnes `output`-Element beschreibt eine Notification. Jedes `operation`-Element ist durch einen eindeutigen Namen gekennzeichnet. Das `message`-Attribut der Elemente `input` und `output` referenzieren jeweils ein zuvor definiertes `message`-Element.

Das Element `binding` definiert die Bindung an ein konkretes Protokoll. Es legt fest, wie genau eine im Element `portType` abstrakt definierte Operation über das Netzwerk transportiert wird. Neben dem Transportprotokoll wird das konkrete Nachrichtenformat festgelegt. Letzteres umfasst den konkreten Aufbau der zu übertragenden Nachricht sowie die Kodierung der Parameter. Für die präzise Beschreibung konkreter Nachrichtenformate wurde WSDL erweitert. So genannte *Binding-Extensions* existieren für SOAP, HTTP und MIME.

Das Element `service` beschreibt einen oder mehrere Ports. Ein Port verknüpft eine Bindung mit einer Netzwerkadresse und definiert so einen Endpunkt. Die zuvor definierte Bindung wird mit dem `port`-Attribut `binding` referenziert. [20]

### *UDDI*

Universal-Description-Discovery-and-Integration (UDDI) stellt eine Spezifikation für den Aufbau eines verteilten Web-Service-Verzeichnisses dar. Die Spezifikation umfasst APIs für das Publizieren und Auffinden von Web Services. Während WSDL die technische Schnittstelle von Web Services beschreibt, beinhaltet UDDI globalere Informationen. Eine UDDI-Beschreibung enthält Informationen über das den Service anbietende Unternehmen. Dazu zählen der Name, Kontaktdaten und Klassifizierungsinformationen wie Branchenangehörigkeit oder geografische Lage. Weiterhin enthält sie Informationen über den angebotenen Service. Neben einer Liste mit Service-Beschreibungen enthält sie auch Klassifizierungsinformationen. Zuletzt enthält eine UDDI-Beschreibung technische Spezifikationen, beispielsweise in Form der Internet-Adresse einer WSDL-Beschreibung. [19]

## **4.) Die Konkurrenz**

Die Idee der Web Services ist nicht neu. Es geht im Wesentlichen darum, Anwendungsfunktionalität, die im Internet vorhanden ist, auf einfache Art und Weise verfügbar zu machen, sodass diese von anderen Anwendungen genutzt werden kann. Im Grunde sind Web Services nichts anderes als eine Prozedur eines Programmes, die auf einen anderen Rechner ausgelagert ist. Die Software, die es überhaupt erst ermöglicht, eine solche entfernte Prozedur aufzurufen, ohne sich um die komplizierten Details der Kommunikation zwischen den beiden Rechnern kümmern zu müssen, wird allgemein als Middleware bezeichnet. Im Gegensatz zu anderen Middleware-Ansätzen realisieren Web Services ein lose gekoppeltes Modell, wobei es sowohl unabhängig von der

Anwendungsplattform und Programmiersprache als auch von Transport- und Nachrichten-Formaten ist. Außerdem bieten Web Services die Möglichkeit, unter Nutzung des HTTP-Protokolls Firewalls zu passieren. Der zu nutzende Dienst kann an einem Ort im Netz aufgerufen werden, welcher zunächst der aufrufenden Anwendung nicht bekannt ist und erst durch einen Dienstvermittler ermittelt wird. Diese Szenarien sind mit den herkömmlichen Middleware-Ansätzen nicht oder nur mit sehr viel Aufwand möglich.

### **5.) Zusammenfassung**

Unter Web Services versteht man sich selbst beschreibende, modulare und autonome Anwendungskomponenten. Diese können als effiziente Methode der verteilten Verarbeitung angewendet oder im Internet publiziert werden. Die im Internet publizierten Web Services können dann von beliebigen Anwendungen jederzeit aufgerufen und genutzt werden.

## 2. 3. Integration von Anwendungen

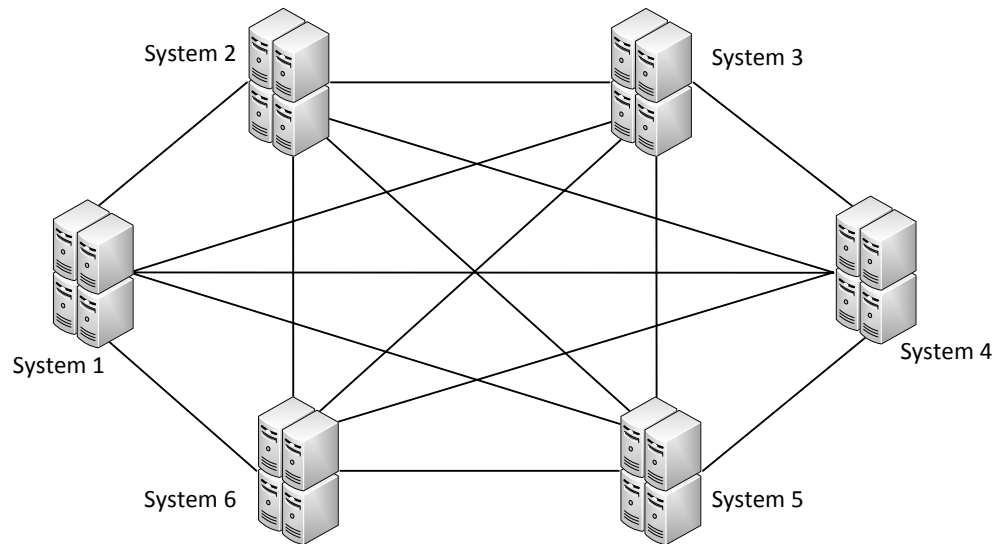
Wie in Kapitel 2.6.1 erläutert, gibt es mehrere Möglichkeiten Anwendungen in ein Portal zu integrieren. Diese Formen der Integration beziehen sich auf die Referenzarchitektur von Portalen [5]. Die meisten Portal-Software-Lösungen am Markt implementieren diese Architektur, sodass die dargelegten Integrationsmethoden angewendet werden können, um verschiedenste heterogene Anwendungen in das eigene Portal zu integrieren. Aber nicht alle Portale nutzen eine vorgefertigte Portal-Software-Lösung oder folgen der Referenzarchitektur. Aus diesem Grund muss sich von den speziellen Portal-Integrationsmethoden abgewandt und die Integration allgemeiner betrachtet werden. Das folgende Kapitel klärt, was allgemein unter Integration von Anwendungen zu verstehen ist und klassifiziert konkrete Integrationstechnologien. Diese Technologien können dann genutzt werden, um Anwendungen beziehungsweise Dienste in ein Portal zu integrieren, die nicht der Referenzarchitektur folgen.

### 2. 3. 1. Integrationstopologien

Hat man mehr als eine Anwendung, die man integrieren möchte, so gibt es mehrere Möglichkeiten diese zu verknüpfen. Dies wird als Integrationstopologie bezeichnet. Es gibt drei verschiedene Topologien für die Integration von Anwendungen. [7]

#### **1.) Punkt-zu-Punkt-Integration**

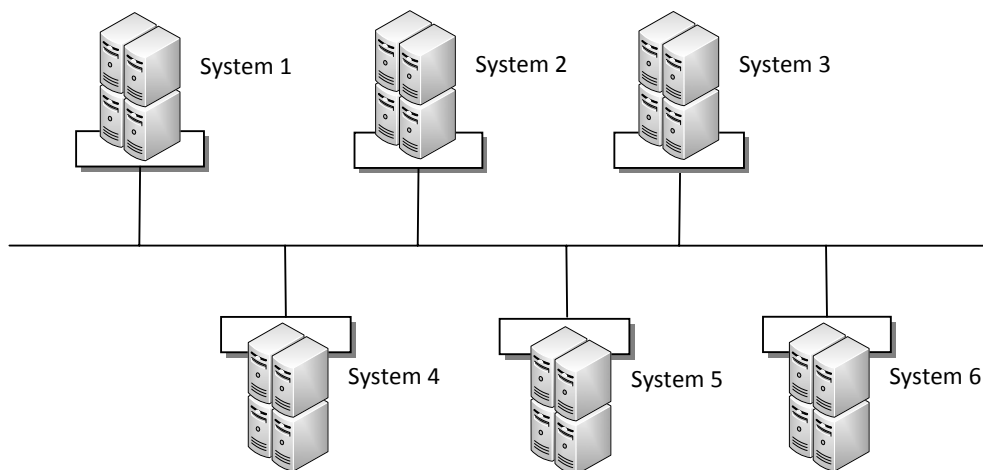
Bei der Punkt-zu-Punkt-Integration werden alle Anwendungen untereinander verbunden. In überschaubaren Integrationsszenarien ist dies einfach zu realisieren. Steigt hingegen die Anzahl der Schnittstellen, so wächst der Wartungs- und Anpassungsaufwand in die Höhe. Zur vollständigen Integration von  $n$  Systemen sind  $(n * (n - 1) / 2)$  Schnittstellen notwendig.



**Abbildung 17: Punkt-zu-Punkt-Integration**

## 2.) Bus-Topologie

Bei der Bus-Topologie ist die Integrationsfunktionalität dezentral durch die angebundenen Einheiten implementiert. Vorteile dieser Topologie sind die niedrige Anzahl an Schnittstellen und die flexible Anpassung an Performanz- und Verfügbarkeitsanforderungen. Nachteilig sind die mögliche Redundanz bei den Infrastrukturfunktionalitäten und der Overhead durch Koordination und Verwaltung der dezentralen Einheiten.



**Abbildung 18: Bus-Topologie**



### 3.) Hub-and-Spoke-Topologie

Bei der Hub-and-Spoke-Topologie stellt ein zentraler Hub die grundlegenden Infrastrukturdienste für die Anwendungsintegration zur Verfügung. Die einzelnen Anwendungen werden mittels standardisierter Schnittstellen über den Hub miteinander verbunden. Vorteil dieser Topologie ist die niedrige Anzahl an Schnittstellen und die Anbindung neuer Systeme mit geringem Aufwand. Nachteilig ist der hohe initiale Aufwand für die Einrichtung der Integrationsinfrastruktur und der Hub, der zum Performanz-Flaschenhals oder *Single-Point-of-Failure* werden könnte.

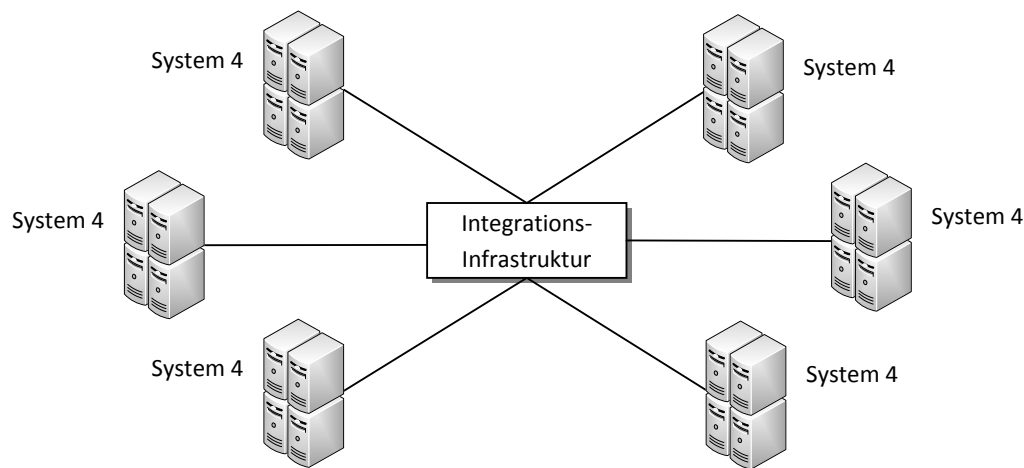


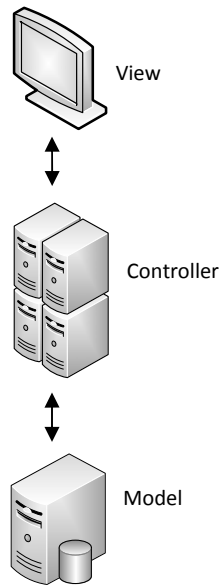
Abbildung 19: Hub-and-Spoke-Topologie

## 2.3.2. Integrationsebenen

Nachdem gezeigt wurde, wie mehrere Anwendungen bei der Integration verknüpft werden können, wird nun auf die einzelnen, zu integrierenden Anwendungen eingegangen. Die Architektur einer Anwendung besteht in den meisten Fällen aus mehreren Schichten. Da auf jeder dieser Schichten eine Integration erfolgen kann, werden diese als Integrationsebenen bezeichnet. Um die verschiedenen Integrationsebenen zu erarbeiten, muss zuerst das Schichtenmodell einer Applikation betrachtet werden.

Das MVC-Modell bezeichnet ein Architekturmuster zur Aufteilung von Applikationen in die drei Einheiten *Datenschicht* (engl.: Model), *Präsentationsschicht* (engl.: View) und

*Verarbeitungsschicht* (engl.: Controller). Ziel des Modells ist ein flexibles Programmdesign, das eine spätere Änderung oder Erweiterung erleichtern und eine Wiederverwendbarkeit der einzelnen Komponenten ermöglichen soll. Außerdem sorgt das Modell bei großen Anwendungen für eine gewisse Übersicht und Ordnung durch Reduzierung der Komplexität.



**Abbildung 20: Aufbau einer Anwendung (MVC-Model)**

Das Model repräsentiert die Daten einer Anwendung. Neben der Verwaltung der Daten führt es alle Änderungen auf diesen aus. Das View übernimmt die grafische Darstellung der Daten. Es stellt eine mögliche visuelle Abbildung des Models dar. Ein View ist dabei immer mit genau einem Model verbunden. Der Controller definiert wie ein Benutzer mit der Applikation interagieren kann. Er nimmt Eingaben vom Benutzer entgegen und bildet diese auf Funktionen des Models oder des Views ab. [21]

Die Integration von Applikationen kann auf jeder dieser Ebenen realisiert werden. Man unterscheidet zwischen der *Datenintegration*, der *Funktionsintegration* und der *Benutzerschnittstellenintegration*. [22]

### 1.) Integration auf Datenebene

Die einfachste Technik zur Integration von Informationssystemen ist auf der Ebene der Daten beziehungsweise auf Basis der Schemata zu Grunde liegender Datenbanken angesiedelt. Das Ziel ist die Sicherstellung anwendungsübergreifender Konsistenz der Daten. Eine erste naheliegende Lösung ist die Einführung eines integrierten Datenschemas. Die Idee ist, nur eine Datenhaltungskomponente und ein Datenschema zu verwenden und Anwendungen so zu entwickeln, dass sie auf dieses Datenschema aufsetzen. Somit können die Konsistenzsicherungsmechanismen des Datenbankmanagementsystems genutzt werden. Dieser Ansatz ist jedoch nur sinnvoll verwendbar, wenn Anwendungen neu entwickelt werden sollen. Die oft vorzufindende heterogene Systemlandschaft macht andere Ansätze erforderlich.

Eine Lösung zur Behebung von Datenheterogenität ist die Transformation in ein gemeinsames Datenmodell zur Beseitigung von Datenmodell-Konflikten (Abbildung 21 links). Eine weitere Lösung ist die Beseitigung von Schemaheterogenität durch Abbildung der unterschiedlichen Schemata auf ein globales Referenzschema (Abbildung 21 rechts). Ebenso könnten Ontologien<sup>1</sup> zur Behandlung semantischer Datenheterogenität genutzt werden, um Synonyme zu entdecken und aufzulösen.

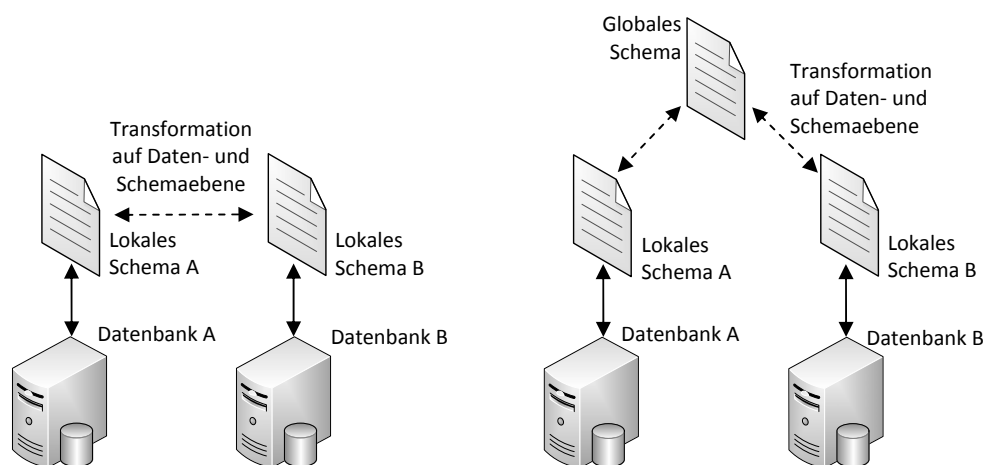
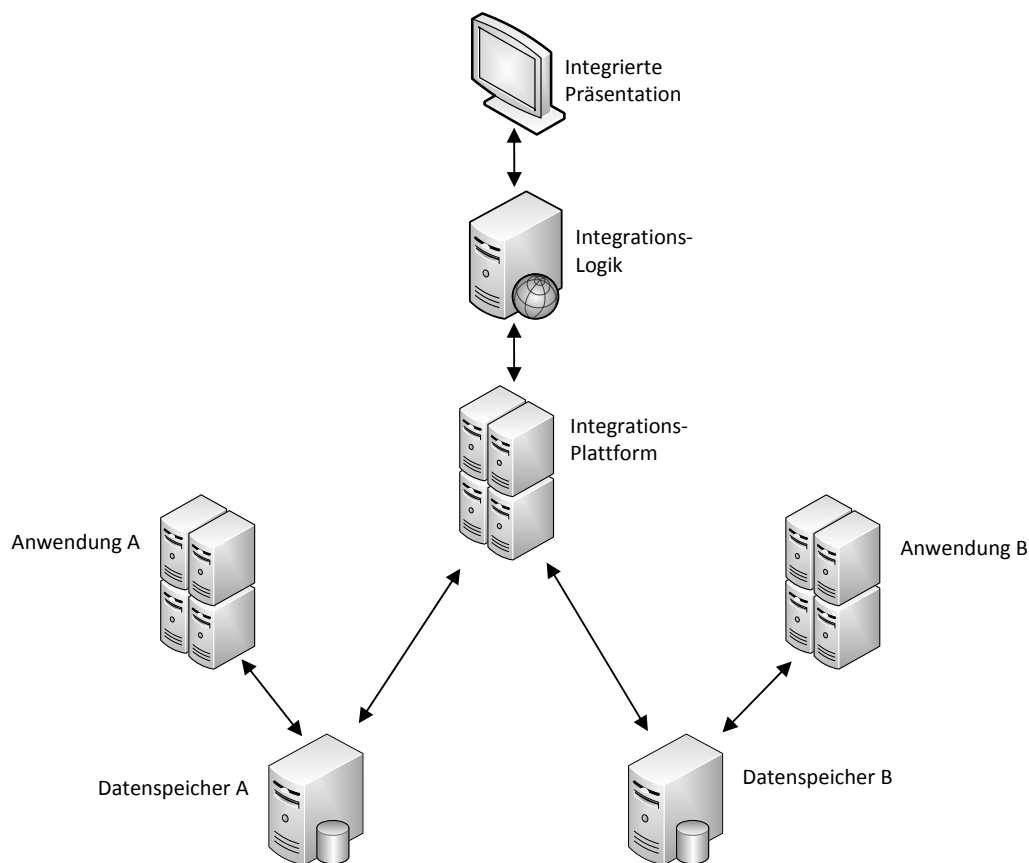


Abbildung 21: Transformationen bei der Datenintegration

<sup>1</sup> Eine Ontologie stellt eine formale Beschreibung der Gegenstände und Beziehungen dar, die für eine Person oder Gruppe von Personen begriffsbindend sind.

Zur Datenintegration wird in heterogenen Systemlandschaften häufig Datenbankzugriffsmiddleware eingesetzt. Sie ermöglicht neben einer für Anwendungen transparenten Einbindung externer Daten den Erhalt vorhandener Zuständigkeiten für Datenpflege und -verwaltung sowie die Wahrung der Autonomie existierender Anwendungen.

Verwendete Technologien sind ODBC<sup>1</sup> [23] und deren Nachfolbertechnologien wie ADO.net [24] und JDBC<sup>2</sup> [25].



**Abbildung 22: Datenintegration**

---

<sup>1</sup> Open-Database-Connectivity

<sup>2</sup> Java-Database-Connectivity; Link: <http://java.sun.com/products/jdbc/>

Die Tabelle 3 stellt die Vor- und Nachteile der Datenintegration gegenüber.

<b><i>Vorteile</i></b>	<b><i>Nachteile</i></b>
<ul style="list-style-type: none"> <li>• Einfach zu realisieren, da keine Anpassung in der Datenstruktur oder in der Anwendungslogik des zu integrierenden Systems erforderlich</li> <li>• Anbindung neuer Systeme auch möglich, wenn kein Quellcode verfügbar</li> </ul>	<ul style="list-style-type: none"> <li>• Semantische Probleme (z.B. Verletzung von Integritätsbedingungen), wenn bei schreibenden Zugriffen auf die Daten eines Systems die in der Anwendungslogik hinterlegten semantischen Informationen ignoriert werden</li> <li>• Keine Nutzung der Anwendungslogik möglich</li> <li>• Änderungen der Datenstruktur eines integrierten Systems führen zu Anpassungen an den Transformationen und dem globalen Datenmodell</li> </ul>

**Tabelle 3: Vor- und Nachteile der Datenintegration**

## ***2.) Integration auf Funktionsebene***

Neben der Integration von Schemata und Daten verschiedener Anwendungen wird jedoch auch die möglichst weitgehende Wiederverwendung und Nutzung von vorhandener Funktionalität aus den integrierten Systemen angestrebt. Die Funktionsintegration ist die semantisch reichhaltigste und flexibelste Art der Anwendungsintegration, denn sie realisiert den Zugriff auf die Geschäftslogik der zu integrierenden Anwendung.

Als Basis für die Systemkopplung dienen Integrationstechnologien, welche die erforderlichen Dienste bereitstellen. Solche Technologien sind Web Services, RPC, RMI, CORBA, EJB, DCOM und MOM. Auf sie wird später noch genauer eingegangen, sie seien hier nur erwähnt.

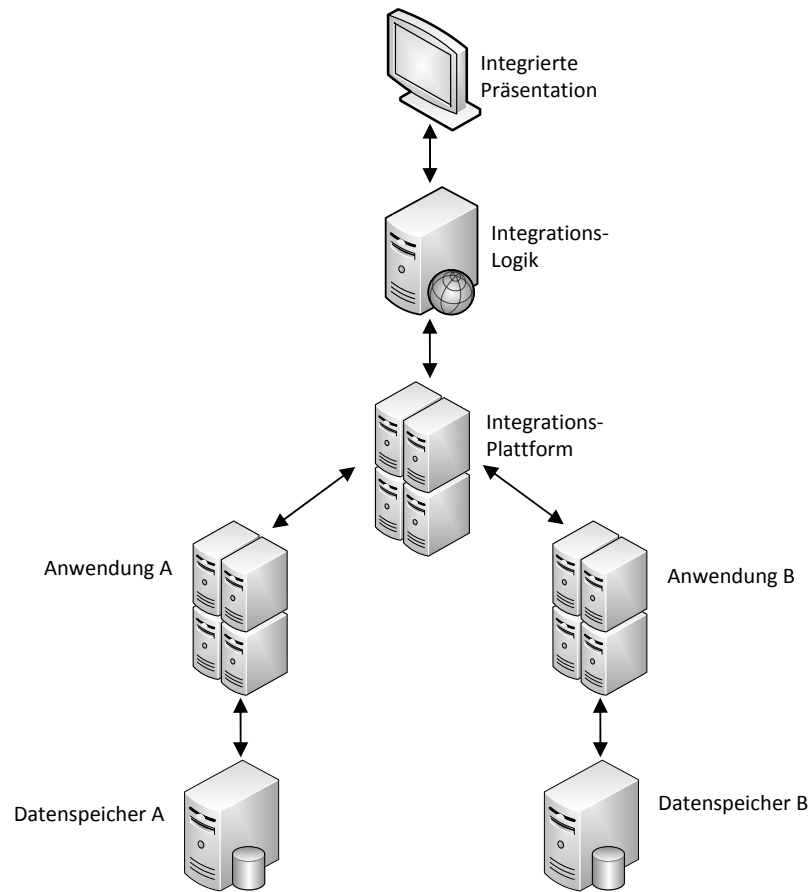


Abbildung 23: Funktionsintegration

Die Tabelle 4 stellt die Vor- und Nachteile der Funktionsintegration gegenüber.

<b>Vorteile</b>	<b>Nachteile</b>
<ul style="list-style-type: none"> <li>• semantisch reichhaltigste Art der Integration</li> <li>• nicht nur Zugriff auf Daten, sondern auch auf Geschäftslogik</li> <li>• Nutzung der Integritäts- und Plausibilitätsprüfungen einer Anwendung vermeidet Integritätsprobleme auch bei schreibenden Zugriffen</li> </ul>	<ul style="list-style-type: none"> <li>• Aufwändig zu realisieren</li> <li>• Unzureichendes Schnittstellenangebot erfordert Anpassung der Anwendung</li> <li>• Schwierig umzusetzen, wenn entsprechende Schnittstellen fehlen oder nicht dokumentiert sind</li> </ul>

Tabelle 4: Vor- und Nachteile der Funktionsintegration

### 3.) Integration auf Präsentationsebene

Neben der Daten- und Funktionsintegration ist die Benutzerschnittstellenintegration die letzte Ebene der Anwendungsintegration. Sie strebt eine einheitliche Benutzeroberfläche an, indem sie die Benutzerschnittstellen der einzelnen Applikationen mittels geeigneter Werkzeuge kombiniert. Diese Form der Integration sollte nur angewendet werden, wenn andere Integrationsansätze aufgrund fehlender Schnittstellen und Anpassungsmöglichkeiten ausscheiden. Zudem bietet sie Nachteile bei der Performanz und eine geringere Flexibilität.

Für die Komposition der verschiedenen Benutzerschnittstellen werden Werkzeuge wie Screen-Scraping [26] und WSRP<sup>1</sup> [6] eingesetzt.

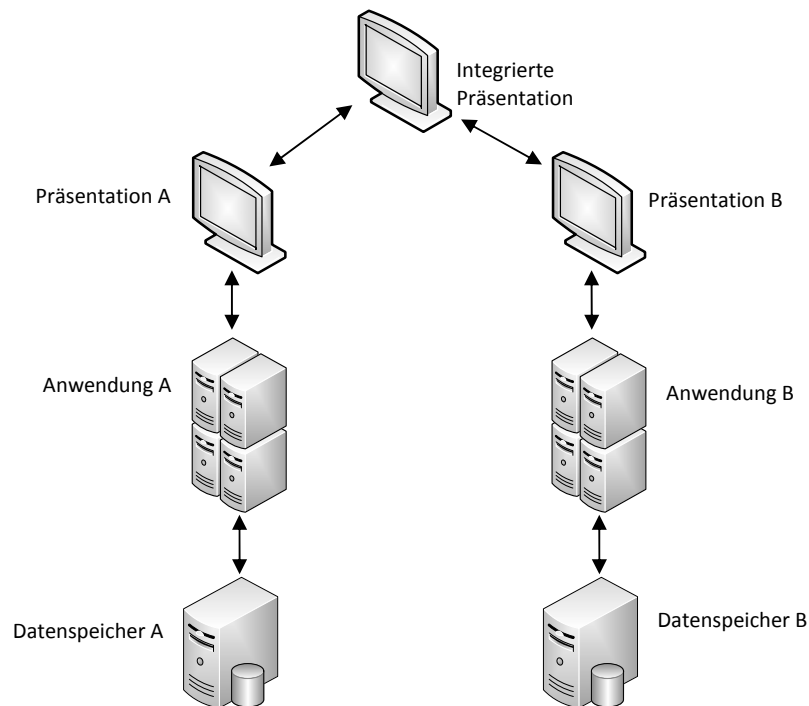


Abbildung 24: Benutzerschnittstellenintegration

<sup>1</sup> Web-Services for Remote-Portlets, Link: [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=wsrp](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrp)

Die Tabelle 5 stellt die Vor- und Nachteile der Benutzerschnittstellenintegration gegenüber.

<b><i>Vorteile</i></b>	<b><i>Nachteile</i></b>
<ul style="list-style-type: none"><li>• Mit geeigneten Werkzeugen schnell und einfach zu realisieren</li><li>• Auch einsetzbar, wenn Anwendung keine API bereitstellt und der Quellcode nicht verfügbar ist</li><li>• In schwierigen Integrationsszenarien oft die einzige Alternative</li></ul>	<ul style="list-style-type: none"><li>• Keine Integration der zu Grunde liegenden Daten und Funktionen</li><li>• Schlechte Performanz und Skalierbarkeit</li><li>• Geringe Flexibilität und Wiederverwendbarkeit</li></ul>

**Tabelle 5: Vor- und Nachteile der Benutzerschnittstellenintegration**



### 2.3.3. Differenzierungsmerkmale

Um die verschiedenen Integrationstechnologien zu vergleichen, sind bestimmte Differenzierungsmerkmale nötig. Dazu zählen das *Kommunikationsmodell*, das *Dienstaufrufmodell* und der *Integrationsgegenstand*.

#### 1.) *Kommunikationsmodell*

Durch das Kommunikationsmodell wird beschrieben, wie die beteiligten Systeme miteinander interagieren. Die beiden folgenden grundsätzlichen Arten der Kommunikation werden unterschieden. [20]

##### *Synchrone Kommunikation*

Unter synchroner Kommunikation versteht man einen Modus der Kommunikation, bei dem der Kommunikationspartner beim Senden oder Empfangen von Daten wartet, bis die Kommunikation abgeschlossen ist. In dieser Zeit ist er blockiert und kann seine Arbeit nicht fortsetzen. Man spricht auch von einer Kopplung des Dienstanutzers mit dem Dienstanbieter. Fast alle interaktiven Anwendungen kommunizieren synchron.

Die wichtigsten Kommunikationsarten bei der synchronen Kommunikation sind *Request/Reply*, *One-way* und *Synchronous-Polling*.

##### *Asynchrone Kommunikation*

Unter asynchroner Kommunikation versteht man einen Modus der Kommunikation, bei dem das Senden und Empfangen von Daten zeitlich versetzt und ohne Blockieren des Prozesses, wie etwa durch Warten auf die Antwort des Empfängers, stattfindet. Asynchrone Kommunikation ist sinnvoll in unzuverlässigen Dienstanbieter-Umgebungen, wenn eine Antwort vom Dienstanbieter nicht sofort oder nicht mit voller Wahrscheinlichkeit erwartet werden kann.

Die wichtigsten Kommunikationsarten bei der asynchronen Kommunikation sind *Message-Passing*, *Publish/Subscribe* und *Broadcast*.

## 2.) Dienstaufmodell

Ein Dienstaufwurf kann sowohl *API-* wie auch *Nachrichten-basiert* erfolgen. [10]

### API-basiert

Eine API ist eine Schnittstelle, die von einem Softwaresystem anderen Programmen zur Anbindung an das System zur Verfügung gestellt wird. Um einen Dienst über eine API zu nutzen, ruft man die Schnittstelle auf und übergibt die notwendigen Daten.

Jede Schnittstelle bezieht sich immer genau auf eine Anwendung, somit ist der Anwendungsaufwurf unmittelbar durch Schnittstellen benannt.

Positiv anzumerken bei APIs ist zum einen der geringe Aufwand bei der Ausführung des Aufrufs und zum anderen die frühe Erkennung von Entwicklungsfehlern bereits beim Kompilieren. Nachteilig sind die geringe Flexibilität, die enge Kopplung und die Abhängigkeit von Änderungen der Schnittstelle.

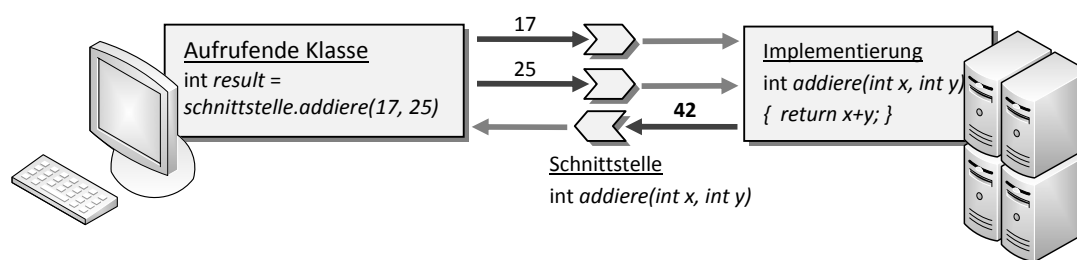


Abbildung 25: API-basierter Dienstaufwurf

### Nachrichten-basiert

Die zweite Möglichkeit einen Dienst aufzurufen sind Nachrichten. Eine Nachricht enthält Informationen über die gewünschte Verarbeitung und die zu verarbeitenden Daten. Der Anwendungsaufwurf ist in der Nachricht integriert. Das Listing 1 zeigt eine einfache Nachricht in einem XML-Format. Diese Nachricht könnte der Input zu einem Dienst darstellen, der zwei Zahlen addiert.

```

<addiere>
  <summand>17</summand>
  <summand>25</summand>
</addiere>

```

Listing 1: Nachricht in XML-Format

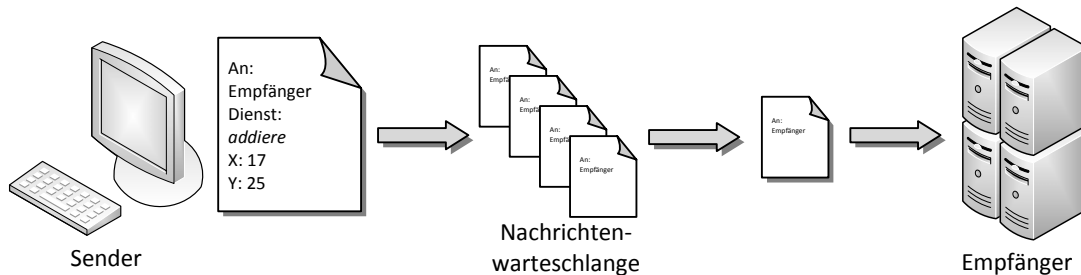


Abbildung 26: Nachrichten-basierter Dienstaufruf

Eine Nachricht ist im Gegensatz zu einer Schnittstelle unabhängig von einer Anwendung, sie hat damit keinen Bezug zu dieser. Ein Dienst muss nicht sofort verfügbar sein, sondern er kann durch eine Nachrichten-Warteschlange die eingehenden Nachrichten sequentiell abarbeiten. Vorteilhaft an Nachrichten als Dienstaufrufmodell sind die lose Kopplung und die hohe Flexibilität sowie die Unabhängigkeit des Dienstes von Änderungen der Nachricht. Negativ anzumerken ist der hohe Aufwand aufgrund der Codierung und Decodierung der Nachricht.

### 3.) Integrationsgegenstand

Ein weiteres Differenzierungsmerkmal ist der Gegenstand der Integration. Dieser kann eine *Funktion*, ein *Objekt* oder ein *Dienst* sein. [27]

#### Funktionsorientierung

Bei dem funktionsorientierten Integrationsparadigma wird über den Aufruf von Prozeduren und Funktionen kommuniziert. Die fehlende Kapselung der Funktionalität bewirkt eine enge Kopplung der integrierten Systeme. Jegliche Änderungen an der Implementierung einer Funktionalität erzwingen die Anpassungen der Klienten, welche diese nutzen.

### Objektorientierung

Bei dem objektorientierten Integrationsparadigma wird die Funktionalität als Methode in verteilten Objekten bereitgestellt. Die wohldefinierten Schnittstellen kapseln den Zugriff auf die Funktionalität. Die Trennung von Schnittstelle und Implementierung entkoppelt die integrierten Systeme.

### Dienstorientierung

Bei dem dienstorientierten Integrationsparadigma sind grobgranulare Dienste durch Schnittstellenbeschreibungen definiert. Dienste können in Verzeichnissen gesucht und zur Laufzeit eingebunden werden. Der Zugriff auf die Dienste erfolgt nachrichten-basiert und lose gekoppelt.

## 2.3.4. Integrationstechnologien

Nachdem erarbeitet wurde, wie verschiedene Anwendungen im Zuge einer Integration verknüpft werden können und an welchen Schichten eine Integration stattfinden kann, wird sich nun konkreten Integrationstechnologien zugewandt. Die nachfolgende Abbildung klassifiziert Integrationstechnologien unter den in Kapitel 2.3.3 erarbeiteten Differenzierungsmerkmalen.

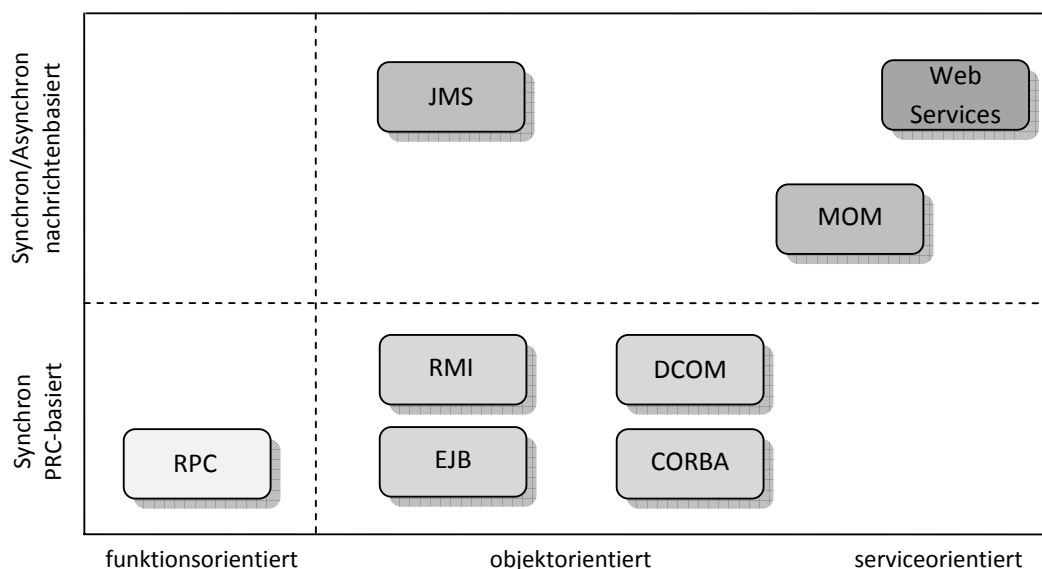


Abbildung 27: Einordnung der Integrationstechnologien

### 1.) Funktionsorientierte und synchrone Integrationstechnologien

*Remote-Procedure-Call* (RPC)<sup>1</sup> [20] ist ein Protokoll zur Implementierung verteilter Anwendungen. Es ermöglicht dem Entwickler den Zugriff auf Funktionen einer entfernten Anwendung unter Abstraktion von Einzelheiten der Netzwerkschicht. RPC arbeitet in der Regel synchron. Ursprünglich wurde das Protokoll von der Firma Xerox entwickelt. Heute existieren eine Reihe von Modellen und Implementierungen, die untereinander inkompatibel sind. Die am weitesten verbreitete Variante ist das *Open-Network-Computing-Remote-Procedure-Call* (ONC-RPC) [20], das vielfach auch als *Sun-RPC* bezeichnet wird. Eine weitere nicht ganz so weit verbreitete Variante ist das *Distributed-Computing-Environment-Remote-Procedure-Call* (DCE-RPC) [20]. Zuletzt sei das so genannte *XML-RPC* [28] erwähnt, welches die versendeten Daten in ein XML-Dokument kapselt und über eine HTTP-Verbindung überträgt.

Nachteilig an RPC ist die Plattform- und Programmiersprachenabhängigkeit. Weiterhin kann durch Ausfall des Empfängers oder Verlust von Nachrichten der Sender blockiert werden, da dieser gemäß der synchronen Kommunikation auf die Antwort wartet.

### 2.) Objektorientierte und synchrone Integrationstechnologien

*Remote-Method-Invocation* (RMI)<sup>2</sup> [20] stellt einen RPC-Mechanismus in Java zur Verfügung. Er ermöglicht im Unterschied zu RPC den Zugriff auf Methoden in entfernten Objekten. Als entfernte Objekte werden Objekte bezeichnet, die sich auf einer anderen Java-Virtual-Machine befinden. Diese kann auf einem anderen Rechner laufen und ermöglicht damit die Kommunikation zwischen verschiedenen Rechnern. Die Plattformunabhängigkeit der Java-Architektur ermöglicht es, auf einfache Weise Programmcode im Netzwerk zu verteilen, auch über verschiedene Plattformen hinweg.

RMI ist zwar plattformunabhängig aber genau wie RPC programmiersprachenabhängig. Ebenso kann durch Ausfall des Empfängers oder Verlust von Nachrichten der Sender blockiert werden. Zudem sind beide Technologien unflexibel, da Adressierungsinformationen des Empfängers im Quellcode des Senders codiert sind und keine Adressierung zur Laufzeit zulassen.

---

<sup>1</sup> RFC: <http://tools.ietf.org/html/rfc1057>

<sup>2</sup> Link: <http://java.sun.com/javase/technologies/core/basic/rmi/>

Die *Common-Object-Request-Broker-Architecture* (CORBA) [20] ist eine Komponentenarchitektur der Object-Management-Group (OMG) und macht Anwendungen als verteilte Objekte zugänglich. Die Kommunikation basiert meist auf RMI und IDL<sup>1</sup> als plattformunabhängige Schnittstellenbeschreibung. CORBA abstrahiert von den Eigenschaften der zu Grunde liegenden Hardwareplattform und Netzwerkinfrastruktur bei entfernten Methodenaufrufen.

Das *Distributed-Component-Object-Model* (DCOM) [20] ist ein von Microsoft definiertes Protokoll, um Programmkomponenten über ein Netzwerk kommunizieren zu lassen. Im Gegensatz zu CORBA verwendet DCOM RPC, um die Kommunikation herzustellen.

Nachteilig an den beiden objektorientierten Technologien ist die durch Objektorientierung eingeschränkte Programmiersprachenunabhängigkeit. Weiterhin können verschiedene CORBA-Implementierungen hinsichtlich der Interoperabilität Probleme machen. Negativ bei DCOM anzumerken ist die unvollständige Plattformunabhängigkeit.

*Enterprise-Java-Beans* (EJB) [29] ist ein Framework zur Entwicklung objektorientierter, komponentenbasierter und verteilter Anwendungen in Java. Sie repräsentieren dabei die Komponenten transaktionsorientierter verteilter Geschäftsprozesse. Der Einsatz von EJB erfolgt im Kontext eines EJB-Containers. Der Container verwaltet die EJB und bietet darüber hinaus Infrastrukturfunktionalitäten wie Namensvergabe, Lebenszyklussteuerung, Persistenz, Messaging, Sicherheit und Transaktionskontrolle. Der EJB-Container ist Teil eines *JavaEE*-Application-Servers. EJBs sind zwar plattform-, aber nicht programmiersprachen-unabhängig. Zudem erlauben sie keine Konfiguration und Integration von Anwendungen zur Laufzeit.

### **3.) Objektorientierte und asynchrone/synchrone Integrationstechnologien**

*Java-Messaging-Service* (JMS) [20] dient dem Austausch von Nachrichten zwischen zwei oder mehreren Clients, die in der Programmiersprache Java geschrieben sind. JMS stellt einen wichtigen Bestandteil von *JavaEE* dar, um asynchrone Kommunikation zu ermöglichen.

*Message-Oriented-Middleware* (MOM) [20] arbeitet nicht mit Methoden- oder Funktionsaufrufen, sondern über den Austausch von Nachrichten. Sie kann sowohl

---

<sup>1</sup> Interface-Definition-Language; Schnittstellenbeschreibungssprache

synchron als auch asynchron arbeiten. Die Nachrichten werden im asynchronen Fall in einer Warteschlange zwischengespeichert und an die betreffende Anwendung weitergeleitet. Der Nachrichtenaustausch ist vollkommen unabhängig sowohl von der Plattform als auch von der Technologie. Die Nachrichten enthalten nicht nur Daten, sondern auch Meta- und Kontrollinformationen. Das Format für die Nachrichten ist nicht festgelegt, in der Praxis hat sich jedoch XML als beliebtes Format etabliert.

Nachteilig an MOMs sind die erhöhte Komplexität beim Programmieren und das Risiko mit dem Ausfall der MOM alle angeschlossenen Systeme lahm zu legen.

#### **4.) Serviceorientierte und asynchrone/synchrone Integrationstechnologien**

Unter Web Services versteht man sich selbst beschreibende, modulare und autonome Anwendungskomponenten. Für detaillierte Informationen sei auf das Kapitel 2.2.6 verwiesen.

### **2.3.5. Zusammenfassung**

Das Kapitel klassifizierte Integrationstechnologien nach vorher erarbeiteten Kriterien. Dabei wurden die konkreten Technologien anhand ihres Kommunikationsmodells, ihres Dienstaufrufmodells und ihres Integrationsgegenstandes in Klassen eingeteilt.

Die erste Klasse ist die der funktionsorientierten und synchronen Integrationstechnologien. Zu dieser Klasse zählt RPC. RMI, CORBA, DCOM und EJBs werden hingegen der Klasse der objektorientierten und synchronen Integrationstechnologien zugeordnet. Nachteilig an beiden Klassen ist die eingeschränkte Programmiersprachen- und Plattformunabhängigkeit. Diese Probleme werden durch die Technologien JMS und MOM gelöst. Sie zählen zu der Klasse der objektorientierten und asynchronen/synchronen Integrationstechnologien. Die letzte Klasse ist die der serviceorientierten und asynchronen/synchronen Integrationstechnologien, zu der Web Services zugeordnet werden.

Alle betrachteten Integrationstechnologien bieten Dienstleistungen zur Vernetzung zwischen Anwendungen an, sodass die Komplexität der zu Grunde liegenden Anwendung und Infrastruktur verborgen wird.





## 3. Das Ilmenauer Alumni-Portal

Dieses Kapitel gibt Aufschluss über das neue Ilmenauer Alumni-Portal. Dazu werden zuerst die Begriffe Alumni, Alumni-Vereinigung und Alumni-Arbeit geklärt, bevor konkrete Alumni-Netzwerke betrachtet werden. Anschließend wird erläutert, warum sich Portale hervorragend für die Alumni-Arbeit eignen. Nach diesen grundsätzlichen Betrachtungen rückt das neue Ilmenauer Alumni-Portal in den Mittelpunkt. Dabei wird verdeutlicht, aus welcher Situation es entstanden ist und wodurch es sich von anderen Absolventen-Portalen abhebt. Nach den konzeptionellen Darstellungen wird die konkrete Umsetzung des Portals beleuchtet. Eine Erörterung, warum das neue Alumni-Portal nicht durch eine Standard-Portal-Lösung realisiert werden sollte, schließt das Kapitel ab.

### 3.1. Alumni, Alumni-Vereinigung

Der Begriff Alumni oder in der Einzahl Alumnus, kommt aus dem Lateinischen und bedeutet wörtlich übersetzt *Zöglinge*. Als Alumni werden heute die Ehemaligen einer Hochschule bezeichnet.

Eine Alumni- oder Absolventen-Vereinigung ist eine Einheit von Personen oder Organisationen, die von einer Hochschule insgesamt oder von einer untergeordneten Hochschulinstitution betrieben wird, um mit den Mitgliedern in regelmäßigem, direktem Kontakt zu stehen und ihnen ein Leistungspaket mit hohem wahrnehmbaren Nutzen anzubieten. Ziel dabei ist die Aktivierung der Mitglieder und der Erhalt der Hochschulbindung durch die Pflege einer intensiven und dauerhaften Beziehung zu ihnen. Absolventen-Vereinigungen beruhen nicht auf Ge-, sondern auf Verbundenheit, da es sich um eine freiwillige Bindung handelt, die auf der Zufriedenheit und wahrgenommenen Vorteilen durch die Alumni basiert. [30]

## **3. 2. Alumni-Arbeit**

Unter Alumni-Arbeit werden alle Aktivitäten zur Kontaktierung und Betreuung von ehemaligen Studierenden verstanden. [31]

### **3. 2. 1. Alumni-Arbeit in den USA**

Mit ihren großen finanziellen Erfolgen durch die Alumni-Vereinigungen und deren gut ausgebildeten Netzwerken gelten die USA als Vorbild für Alumni-Arbeit. Unterstützung durch die Alumni, ob in Form von Geld, Sachmitteln oder Dienstleistungen, ist dort längst selbstverständlich und aus der Universitätskultur nicht mehr wegzudenken. Einige amerikanische Hochschulen können sogar ein Drittel ihres Etats aus der Unterstützung, die sie von Ehemaligen erhalten, decken.

In den USA gibt es Alumni-Vereinigungen bereits seit 200 Jahren. Sie gehören zur Geschichte der US-Universitäten dazu und haben bereits professionellen Charakter.

### **3. 2. 2. Alumni-Arbeit in Deutschland**

In Deutschland begannen die Alumni-Aktivitäten erst vor etwa 20 Jahren und bleiben auch heute noch hinter dem Vorbild der USA zurück. Zurzeit gibt es zirka 700 Alumni-Vereinigungen im deutschsprachigen Raum, also in Deutschland, Österreich und der Schweiz. [31] Die Führung der Vereine ist im Gegensatz zu den USA kein Vollzeitberuf, der von professionellen Führungskräften ausgeübt wird.

Der große Unterschied zu der Alumni-Arbeit in den USA liegt in kulturellen Aspekten begründet, wie etwa die Bereitwilligkeit zu ehrenamtlichen Tätigkeiten, der große Willen zum Mitgestalten, die hohe Anerkennung des sozialen Engagements und dem hohen Sozialprestige der universitären Ausbildung in der Gesellschaft. [32] Die Ehemaligen sind stolz darauf, der Universität ehrenamtlich und unentgeltlich helfen zu können, weil die Zeit an der Universität als ein wichtiger Lebensabschnitt betrachtet wird, für den sich die Absolventen bei ihrer Universität bedanken möchten.

Um die Alumni-Arbeit professionell durchführen zu können, müssen in Deutschland einige Versäumnisse aufgeholt werden. Die an vielen Hochschulen fehlende Hochschulbindungsarbeit sowohl während des Studiums als auch nach Studienende, bietet für die Bindung der Absolventen an die Hochschule eine schlechte Grundlage. Da sich die Studierenden und späteren Ehemaligen nicht ausreichend mit ihrer Hochschule identifizieren, sinkt auch die Bereitschaft, sich nach dem Studium für diese einzusetzen. Weiterhin ist die Alumni-Arbeit als Teil der Hochschulstrategie noch in der Entstehungsphase. Es fehlen die Adressen der Absolventen sowie ihre Präferenzen und Einstellungen zur Hochschule. Weitere Probleme ergeben sich für die Entwicklung von Alumni-Netzwerken in Deutschland auch durch die historische Organisationsentwicklung deutscher Universitäten. Hochschulen sind demokratisch organisiert, was zu längeren Wegen in der Entscheidungsfindung führt und Auswirkungen auf die strategische Universitätsentwicklung hinsichtlich der Einführung von Alumni-Netzwerken hat. Hinzu kommt der Bildungsauftrag der Universität. Die Gesellschaft hat Anforderungen an Forschung und Lehre, das bedeutet neben den Anspruchsgruppen der Studenten, Absolventen und Drittmittelgeber gibt es eine Reihe unterschiedlicher Interessengruppen, an deren Wünschen sich die Universität ebenfalls orientieren muss. Ein weiteres Hindernis ist die Gesetzgebung bezüglich des Datenschutzes. Aus den Datenschutzbestimmungen ergibt sich, betrachtet man das Alumni-Netzwerk als von der Universitätsverwaltung unabhängige, also als nicht-öffentliche Einrichtung, dass die Daten der Alumni durch die Universitätsverwaltung nicht ohne deren Einwilligung an die Alumni-Organisation weitergeleitet werden können. Somit wird die für die Alumni-Arbeit so wichtige Informationsgewinnung erschwert.

In der Entwicklung der deutschen Alumni-Arbeit sind aber auch wichtige Eckpunkte zu nennen, die eine positive Entwicklung deutlich machen. Zunächst handelt es sich dabei um die 1997 ausgesprochene Empfehlung der Hochschulrektorenkonferenz (HRK) zur Gründung von Absolventen-Vereinigungen und so genannten Career-Services sowie eine aktive Bindungsarbeit bereits während des Studiums. Diese Empfehlung geht auf die immer höhere Bedeutung des Erfolgs der Alumni für das Image der Hochschule zurück. Ein weiterer wichtiger Punkt ist die Gründung des Verbandes deutschsprachiger Alumni-Organisationen *alumni-clubs.net e.V.* im Jahr 1997, dessen zentrale Aufgabe die Unterstützung der Alumni-Arbeit ist. Dazu gehört ein Informations- und Erfahrungsaustausch, Hilfestellung beim Aufbau von Alumni-Projekten und -Initiativen, Öffentlichkeitsarbeit für die Alumni-Arbeit sowie die Sammlung von Informationen zum

Thema Alumni-Arbeit in einer Wissenschaftsdatenbank, damit diese zentral und für alle verfügbar sind. Weiterhin beteiligt sich der Verein an wissenschaftlichen Arbeiten und Forschung zum Thema Alumni und führt eigene Analysen und Studien durch. [32]

## **3.3. Alumni-Netzwerke**

Nachdem die Unterschiede der Alumni-Arbeit in den USA und in Deutschland deutlich gemacht wurden, werden nun konkrete Alumni-Netzwerke betrachtet. Zuerst das Absolventen-Netzwerk der TU Ilmenau und im Anschluss Alumni-Netzwerke anderer Hochschulen.

### **3.3.1. Absolventen-Netzwerk der TU Ilmenau**

Das Alumni-Netzwerk der TU Ilmenau dient dem Zweck der Kontaktaufnahme und Kontaktpflege sowie dem Angebot diverser Dienste für die Mitglieder, um diese zu aktivieren und die Bindung der Ehemaligen an die Universität zu erhalten. Es wurde von der Universitätsleitung initiiert, besteht seit dem Jahr 2001 und führt bereits mehr als 6300 Alumni in ihrer Datenbank<sup>1</sup>.

Das Absolventen-Netzwerk der TU Ilmenau ist unterteilt in einen nationalen und einen internationalen Teil. Der internationale Teil enthält Informationen, die besonders für ausländische Absolventen interessant sind, etwa Veranstaltungen des Akademischen Auslandsamtes oder Kontaktdaten zu ausländischen Alumni-Vereinigungen der TU-Ilmenau. Zu den Angeboten des nationalen Teils des Absolventen-Netzwerkes zählen die Absolvententage, die Alumni-Webpräsenz auf der Web-Seite der Universität sowie der jährliche Newsletter mit Informationen zu Veranstaltungen und Höhepunkten im Universitätsleben. Zudem ist es möglich, über die Alumni-Verantwortlichen Kontakt zu anderen Absolventen herzustellen und sich bei der Organisation von Ehemaligen-Treffen unterstützen zu lassen.

Viele Dienste des Absolventen-Netzwerkes der TU Ilmenau sind online über die Alumni-Webpräsenz<sup>2</sup> verfügbar. Diese Web-Seite ist die Schnittstelle zwischen den Alumni und

---

<sup>1</sup> Quelle: Absolventen-Netzwerk, Monika Schoß [Stand: April 2007]

<sup>2</sup> Link: <http://www.tu-ilmenau.de/uni/Alumni.51.0.html>

dem Absolventen-Netzwerk der Universität und bietet den Zugang zu den angebotenen Alumni-Diensten. Solche Dienste sind der Career-Service, in Form einer Job- und Praktikumsbörse sowie Weiterbildungen, die aber von der Service GmbH der Universität organisiert werden. Weiterhin werden Informationen zu Absolventen- und Kontaktmessen sowie Veranstaltungen angeboten.

### 3.3.2. Absolventen-Netzwerke anderer Hochschulen

Die meisten Alumni-Netzwerke versuchen ihre Dienste online verfügbar zu machen. Der Grund ist das Internet, als Kommunikationsmedium Nummer eins. Da alle Alumni regional verteilt sind, ist das Internet die beste Möglichkeit trotzdem die Dienste der Absolventen-Vereinigung in Anspruch zu nehmen. Aus diesem Grund spiegeln die meisten Alumni-Web-Seiten die Dienste der Alumni-Netzwerke der jeweiligen Hochschulen wieder.

Eine Recherche und Analyse der Alumni-Web-Seiten anderer Hochschulen ergab ein ähnliches Portfolio an Funktionalität und Umfang im Vergleich mit der Alumni-Web-Seite der TU Ilmenau. Das Angebot reicht von Informationen über Veranstaltungen für Alumni an Ihrer Universität und anderen Terminen bis hin zu Angeboten für Beratungen zum Berufseinstieg. Meist sind auch eine Jobbörse und eine Suchfunktionalität realisiert. Damit ordnet sich das Angebot der Ilmenauer Alumni-Arbeit im deutschen Mittelmaß ein. Es gibt zwar wesentlich bessere Angebote, aber auch wesentlich schlechtere oder keine Alumni-Arbeit an deutschen Hochschulen.

## 3.4. Alumni-Portale

Wie in der Definition nach [30] angemerkt, beruht eine Alumni-Vereinigung nicht auf Ge-, sondern auf Verbundenheit, die auf der Zufriedenheit und den wahrgenommenen Vorteilen durch die Alumni basiert. Der Autor ist überzeugt, dass bei Vorhandensein eines „[...] *Leistungspaketes mit hohem wahrnehmbaren Nutzen* [...]“ seitens der Hochschule eine „[...] *intensive[n] und dauerhafte[n] Beziehung* [...]“ zwischen den Alumni entsteht. Ein solches Leistungspaket kann ein Portal bieten.

Ein Portal ist eine Art Zugangspunkt, über den ein Nutzer eines Systems alle für ihn relevanten Informationen und Anwendungen angeboten bekommt, um einer bestimmten Aufgabe oder einem Wunsch nachzukommen. Das Alumni-Portal ist im weiteren Sinne das Tor zur Alumni-Arbeit, das den Alumni alle online verfügbaren Dienste des Alumni-Netzwerkes anbietet. Portale schaffen einen zentralen, einheitlichen und systemunabhängigen Zugang zu Informationen und Diensten (vgl. Kapitel 2.1).

Alumni-Portale vereinfachen zudem die Kommunikation zwischen den Mitgliedern und sorgen für mehr Gemeinschaft. Sowohl persönliche Kontaktdaten als auch Änderungen der Kontaktdaten können den ehemaligen Kommilitonen mitgeteilt werden, ohne explizit Institutionen zu bemühen. Weiterhin ermöglichen sie eine effizientere Distribution von Informationen an die Mitglieder und die Bereitstellung eines synergetischen Wissenspools, etwa durch Publikationen in Form von Diplomarbeiten oder wissenschaftlichen Arbeiten. Für die Hochschule bringt ein Alumni-Portal eine starke Verringerung des Aufwands für die Mitgliedsverwaltung, denn Alumni können sich selber anmelden, ihre persönlichen Daten verwalten, ändern und löschen. Nicht zu vernachlässigen ist die Funktion des Portals als eine Community. Sie stärkt den Zusammenhalt und die Gemeinschaft der Alumni untereinander und festigt die Bindung zu der Hochschule. Diese Identifizierung des Alumni mit der Hochschule ist elementar wichtig, denn sie hat einen großen Einfluss auf das Image und somit auf den Erfolg einer Hochschule.

Zusammenfassend ist zu sagen, dass sich Portale bestens für die Alumni-Arbeit eignen, denn sie bieten einen zentralen und einheitlichen Zugangspunkt, über den die Alumni alle relevanten Informationen und Dienste angeboten bekommen. Zudem stärkt eine solche Gemeinschaft die Bindung der Alumni untereinander als auch zu der Hochschule.

## 3. 5. Konzeption des Ilmenauer Alumni-Portals

Im vorherigen Kapitel wurde die aktuelle Ilmenauer Alumni-Arbeit ausführlich analysiert mit der Erkenntnis, dass es zwar den Großteil der Dienste realisiert, die andere Alumni-Vereinigungen auch bieten, doch von einem umfangreichen Portfolio an Diensten kann nicht die Rede sein. Im Gegensatz dazu gibt es Alumni-Vereinigungen anderer Hochschulen, die sich mit innovativen Ideen und Lösungen vom Rest abheben. Die Ilmenauer Alumni-Arbeit ist dadurch im Moment nur Mittelmaß.

### 3. 5. 1. Handlungsempfehlungen zukünftiger Alumni-Arbeit

Eine Ilmenauer Studentin der angewandten Medienwissenschaft beschäftigte sich in ihrer Diplomarbeit mit dem Thema „*Alumni-Arbeit an deutschen Hochschulen*“ [32] und untersuchte dabei auch die Arbeit an der TU Ilmenau. Ihre Forschungsfrage „*Wie sollte die Alumni-Arbeit an der TU Ilmenau bezüglich der Angaben der Ilmenauer Alumni zu möglichen Leistungen und Angeboten des Alumni-Netzwerkes in Zukunft gestaltet werden?*“ stellte sie mehr als 6300 Ilmenauer Absolventen. Aus dieser Befragung und den Ergebnissen ihrer Arbeit schlussfolgerte sie konkrete Handlungsempfehlungen hinsichtlich der Gestaltung der zukünftigen Alumni-Arbeit an der TU Ilmenau. Die wichtigsten sollen hier kurz genannt sein:

- Zielgruppenspezifische Alumni-Strategie (Entwicklung einer konkreten Alumni-Strategie mit definierten Zielgruppen, Zielen und eine klare Herausstellung der Leistungen des Alumni-Netzwerkes für alle Zielgruppen)
- Erneuerung der vorhandenen Software und Datenbank (Sammlung von Informationen über die Alumni und deren Verwaltung)
- Kontakte zu anderen Alumni-Netzwerken (neue Aspekte, Ideen und Anregungen für die mögliche Ausgestaltung des Alumni-Netzwerkes erhalten)

- Frühzeitige Information (zukünftigen Absolventen sollten schon vor dem Studium, spätestens aber während des Studiums auf das Alumni-Netzwerk aufmerksam gemacht werden über die konkrete Angebote und Leistungen des Netzwerkes für die Studierenden)
- Kommunikationsmöglichkeiten der Alumni untereinander (Zur Unterstützung bei der Organisation von Ehemaligentreffen und die Vermittlung von Kontakten zu anderen Alumni)
- Konkrete neue Angebote (Einladungen zu Universitätsfesten, wissenschaftliche Fachveranstaltungen und Vorträge sowie Weiterbildungsangebote an der TU Ilmenau, Absolventen-Jahgangsbuches, Einrichtung einer lebenslangen Alumni-Email-Adresse)
- Die Ausweitung des Internetangebots (Möglichkeit des Zugriffs auf Publikationen der Universität, Mitgliederverzeichnis, Suchfunktion nach Alumni)
- Informationen über die Universität (Rankings, Forschungsprojekte, öffentliche Fachveranstaltungen und Vorträge, wissenschaftliche Kongresse und Tagungen, Universitätsfeste) und über die Ehemaligen (Branchenprobleme und Erfolge, Aktivitäten der Ehemaligen in einzelnen Fachbereichen)
- Kommunikationsmedium Internet (Das Internet als Kommunikationsmedium bietet die besten Voraussetzungen mit allen Alumni Kontakt aufnehmen zu können)

### 3. 5. 2. Das Projekt AWWP

Parallel zu der obigen Arbeit entstand im Fachgebiet Integrierte Hard- und Software-Systeme der Fakultät für Informatik und Automatisierung ein zweijähriges Projekt zum Ausbau des Alumni-Portals der TU Ilmenau und der Entwicklung eines neuen fachlichen Alumni Weiterbildungswebportals. Hervorgebracht durch eine ausführliche Analyse und enge Zusammenarbeit mit dem Alumni-Netzwerk der Universität wurden folgende Ziele an ein zukünftiges Alumni-Portal gestellt:

- Präsentation der Alumni durch Lebenslauf und Veröffentlichungen
- Einbindung der Jobbörse der TU Ilmenau
- Neue Recherchemöglichkeiten
  - Bildungsportal Thüringen, Bibliothek der TU Ilmenau / Ilmedia



- Ausleihmöglichkeiten von der Bibliothek der TU Ilmenau
- Weiterbildungswebportal
  - E-Learning, Online Weiterbildungsmöglichkeiten
- Neue webbasierte Fachkurse
- Fachliche Newsletter
- Vernetzung mit Alumni Portalen anderer Hochschulen

### ***1.) Leistungsumfang des neuen Alumni-Portals***

Die Absolventen der TU Ilmenau sollen durch das geplante Alumni-Portal die Möglichkeit bekommen, sich den anderen Alumni oder externen Besuchern des Portals zu präsentieren. Um dieser Aufgabe nachzukommen, können sie einen Lebenslauf pflegen und wissenschaftliche Arbeiten über dieses Portal publizieren. Die Jobbörse der TU Ilmenau soll ebenfalls in das Portal integriert werden, denn zu einem professionell gestalteten Alumni-Netzwerk gehören die Begleitung der Karriere und die Erleichterung des Einstiegs in das Berufsleben. Weiterhin bietet das zukünftige Absolventen-Portal neue Recherchemöglichkeiten, denn es vereint die Suche über verschiedene, dem Portal angebundene Informationsquellen. So ist es möglich, mit einer einzigen Eingabe eine Suche zu initiieren, die sowohl das Bildungsportal Thüringen als auch die Bibliothek der Universität und Universitäts-Seiten selbst abdeckt. Eine Ausleihmöglichkeit der Bibliothek rundet die Recherche ab. Ein weiterer wichtiger Dienst in dem neuen Ilmenauer Alumni-Portal sind die Weiterbildungsangebote. Mit diesen Angeboten kann dem wachsenden gesellschaftlichen Anspruch nach lebenslangem Lernen entsprochen werden. Dies kann sowohl vor Ort durch die vorhandenen Kapazitäten der Hochschule beziehungsweise der Fakultäten als auch in Form von Online-Angebote erfolgen. E-Learning- und Online-Weiterbildungs-Material sind im Bereich der Alumni-Arbeit aufgrund der örtlichen Entfernungen besonders sinnvoll. Ein fachlicher Newsletter mit Informationen über aktuelle Forschungsprojekte, öffentliche Fachveranstaltungen und Vorträge, wissenschaftliche Kongresse und Tagungen soll das Programm ergänzen. Auch über Veranstaltungen der Alumni-Vereinigungen soll informiert werden, denn sie sind ein wichtiges Element der Kontaktaufnahme, Kontaktpflege und Vernetzung sowohl der Alumni untereinander als auch zur Hochschule.

Bei genauerer Betrachtung der Ziele an ein zukünftiges Absolventen-Portal und den Handlungsempfehlungen hinsichtlich der Gestaltung der zukünftigen Alumni-Arbeit (vgl. Kapitel 3.5.1) sind viele Gemeinsamkeiten zu erkennen. Die Handlungsempfehlungen, die durch das Portal realisiert werden, sind unterstrichen und zeigen auf, dass mit der Realisierung dieses Portals ein richtiger Weg eingeschlagen wurde, um die Alumni-Arbeit an der TU Ilmenau voranzutreiben.

#### ***2.) Partneruniversitäten***

Alle oben genannten Ziele sollen in einem neuen Absolventen-Portal vereint werden und so die Ilmenauer Alumni-Arbeit fördern. Dieses System wird später nicht nur von der TU Ilmenau genutzt, sondern auch von den Universitäten der internationalen Partner des Projekts. Das Alumni-Portal wird zusätzlich an den Universitäten Al-Baath und Tichrin in Syrien, an der Universität Jijel in Algerien und an der Suez-Canal Universität in Ägypten eingesetzt werden. Damit trägt dieses Projekt nicht nur zur Bindung ausländischer Alumni an Deutschland bei, sondern liefert auch einen großen Beitrag zur Entwicklung eines hochschulweiten Alumni-Konzeptes. National wird das Projekt mit dem Akademische Auslandsamt der TU Ilmenau, der Bibliothek und dem Rechenzentrum der TU Ilmenau sowie dem Bildungsportal Thüringen abgestimmt.

#### ***3.) Dienste des zukünftigen Alumni-Portals***

Nachfolgende Abbildung verdeutlicht das Zusammenspiel der Dienste des Portals. Das Alumni-Portal ist die Schnittstelle zur Alumni-Arbeit, welches den Alumni alle online verfügbaren Dienste des Alumni-Netzwerkes über einen zentralen und einheitlichen Zugang anbietet.

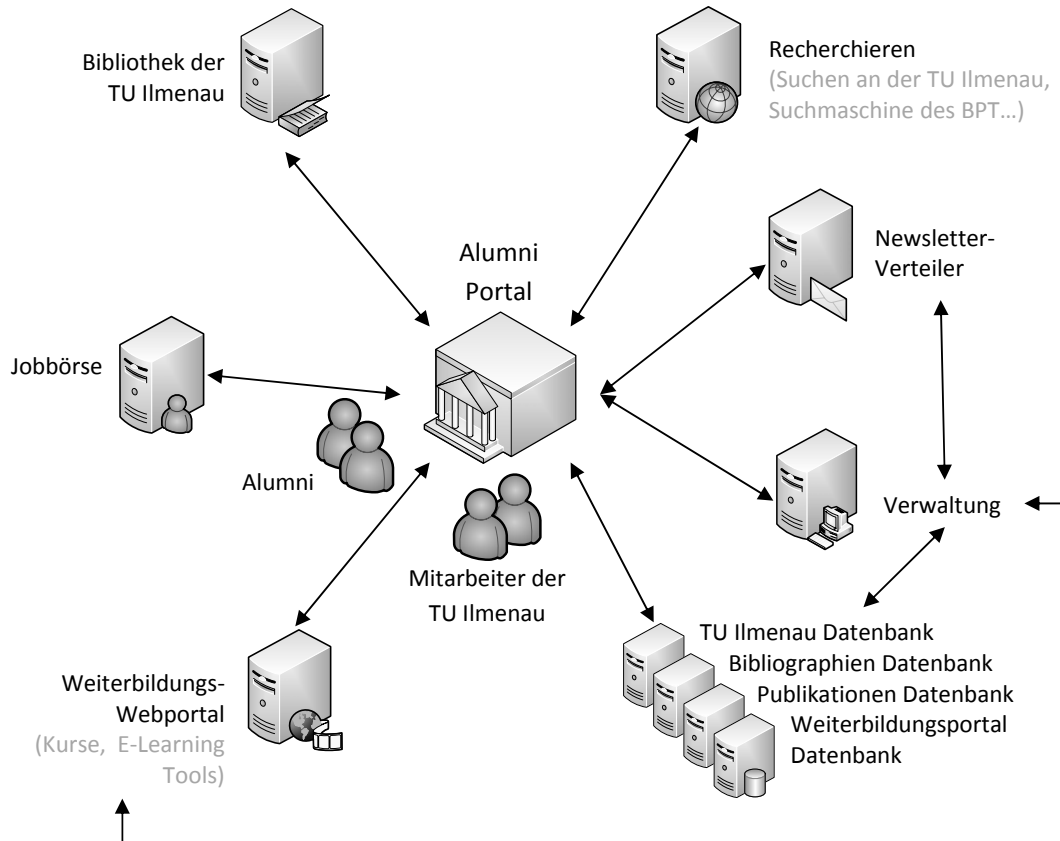


Abbildung 28: Zusammenspiel der Dienste des Alumni-Portals

### 3.5.3. Herausstellungsmerkmale zu anderen Alumni-Portalen

Betrachtet man die obigen Ziele an ein neues Ilmenauer Alumni-Portal und hat dabei einen Blick auf die Alumni-Portale der anderen Hochschulen, so setzt sich das Absolventen-Portal der TU Ilmenau vom Rest ab.

Während die meisten Alumni-Portale lediglich über News und Events informieren, Stellenbörsen präsentieren, Hilfe beim Berufseinstieg geben und die Möglichkeit bieten, ehemalige Kommilitonen zu suchen, bietet das Ilmenauer Portal einiges mehr. Nicht allein der Umfang an Diensten spielt dabei die entscheidende Rolle, sondern eher die Stellung des einzelnen Alumni zu dem Portal. Die meisten Alumni-Portale, wenn man von Portalen reden kann, lassen die Alumni sehr passiv wirken. Die Alumni-Vereine der einzelnen

Hochschulen offerieren Angebote, mit denen sich die Alumni über aktuelle Geschehnisse an der jeweiligen Hochschule informieren können. Die Informationen fließen nur in eine Richtung. Der einzelne Alumni kann nicht aktiv mitwirken, indem er sein Wissen und seine Erfahrungen einbringen kann, um es anderen Alumni anzubieten (vgl. Kapitel 2.2.4). Der Erfolg einer Hochschule ist von seinen Absolventen abhängig. Die positiven Erfahrungen und Erfolge der Absolventen nach dem Studium reflektieren auf das Image der Hochschule und steigern deren Ansehen.

Das neue Ilmenauer Alumni-Portal bietet den Alumni die Möglichkeit ihr Wissen und ihre Erfahrungen anderen Alumni mitzuteilen<sup>1</sup>. Die von den Alumni in dem Portal publizierten wissenschaftlichen Arbeiten steigern nicht nur die Reputation der Alumni, sondern auch die der Hochschule.

Weiterhin bietet das neue Absolventen-Portal der TU Ilmenau die Möglichkeit der Personalisierung der Alumni an. Ein Alumni kann Lebensläufe pflegen und diese anderen mitteilen. Lebensläufe können sowohl Nutzern des Portals als auch externen präsentiert werden. Der Alumni hat die volle Kontrolle über die Sichtbarkeit seiner gemachten Eingaben.

Die Möglichkeit der Einbringung von Wissen und die Personalisierung sind zwei wichtige Punkte, die ein Portal ausmachen. Ein Portal lebt von seinen Diensten und seinen Nutzern. Sie machen ein Portal erst erfolgreich und lebendig.

Viele Alumni-Portale bezeichnen sich zu Unrecht als Portale, denn sie bieten nicht die notwendigen Voraussetzungen eines Portals (vgl. Kapitel 2.1.3). Sie sind eher eine einfache und überschaubare Linksammlung als ein personalisierter Zugang zu Diensten und Informationen.

---

<sup>1</sup> Auch als *Social Networking* bezeichnet, Social Networking ist ein Phänomen des Web 2.0 [49]

## 3. 6. Realisierung des Ilmenauer Alumni-Portals

Nachdem im vorherigen Kapitel das Konzept des neuen Ilmenauer Alumni-Portals beleuchtet wurde, wird sich nun der konkreten Realisierung zugewandt.

### 3. 6. 1. Portal-Software-Lösungen

Für die Erstellung von Portalen werden am Markt verschiedene Software-Lösungen angeboten. Diese Lösungen bieten schon eine Vielzahl an Funktionen, die genutzt werden können, um daraus ein eigenes Portal zu entwickeln. Dabei kann auf Funktionen wie die Erstellung von Frontends mit Portlets, die Integration vorhandene Systeme, die Personalisierung, diverse Suchfunktionen, ein Dokumenten- und Content-Management, sowie ein Business-Prozess-Management zurückgegriffen werden.

Nach der Gartner Group<sup>1</sup> lässt sich der Portal-Software-Markt, abhängig von der Marktpräsenz und Abdeckungsgrad, in vier Quadranten einteilen. Marktführer von Portal-Software ist IBM, gefolgt von SAP, Oracle, BEA und Microsoft. Diese sind laut Gartner jene Hersteller, denen es gelingt, die Bedürfnisse des Marktes heute und auch zukünftig zu befriedigen. Ihre Produkte sind hochgradig integriert und skalierbar.

---

<sup>1</sup> Link: <http://www.gartner.com>

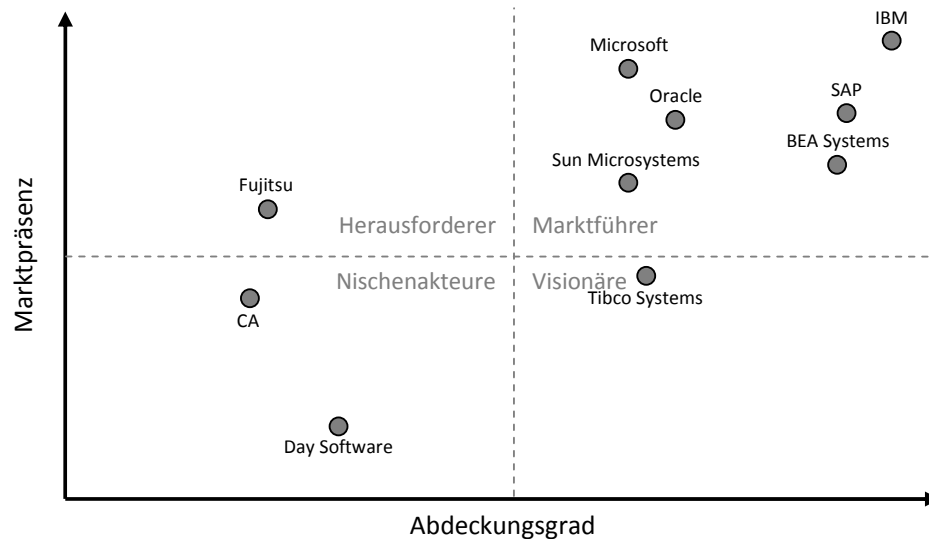


Abbildung 29: Markteinteilung nach Gartner (Quelle: [33])

Die Einführung einer Portal-Software-Lösung erfordert eine ausführliche Vorbereitung und eine zielgerichtete Durchführung. Hierfür existieren am Markt verschiedene Methoden. Die bekannteste Methode ist die Portal-Design-und-Analyse-Methode (PADEM)<sup>1</sup> [1] vom Fraunhofer Institut-für-Arbeitswirtschaft-und-Organisation (IAO). Neben einer Definition des Vorgehens nach einzelnen Projektphasen, existieren für jede dieser Phasen Dokumente wie Verfahrenshinweise, Checklisten und Fragebögen, die die Durchführung zielorientiert gestalten und damit die Komplexität begrenzen.

Die meisten Portal-Software-Lösungen sind auf Unternehmens-Portale ausgerichtet.

### 3. 6. 2. Unternehmens-Portale

Bereits Ende der 90er Jahre wurde durch Forschung und Wirtschaft erkannt, dass die erfolgreiche Umsetzung des elektronischen Geschäftsverkehrs in und zwischen Unternehmen von dem Einsatz einer übergreifenden Plattform zur Abbildung von Geschäftsprozessen abhängt. Diese Anforderungen erfüllen Unternehmens-Portale. Ein Unternehmens-Portal ist eine Informations-, Kommunikations- und Arbeitsplattform, die verschiedene Anwendungen integriert und in der den Nutzern die Inhalte personalisiert und anforderungsgerecht verfügbar gemacht werden.

<sup>1</sup> Link: <http://www.padem.iao.fraunhofer.de/>

Zielsetzung von Unternehmens-Portalen ist die Bereitstellung und Unterstützung von Prozessen zwischen dem Unternehmen und den Nutzern (Abbildung 30). Die Nutzer eines Unternehmens-Portal sind Mitarbeiter, Kunden, Lieferanten oder andere Kooperationspartner des betreffenden Unternehmens.

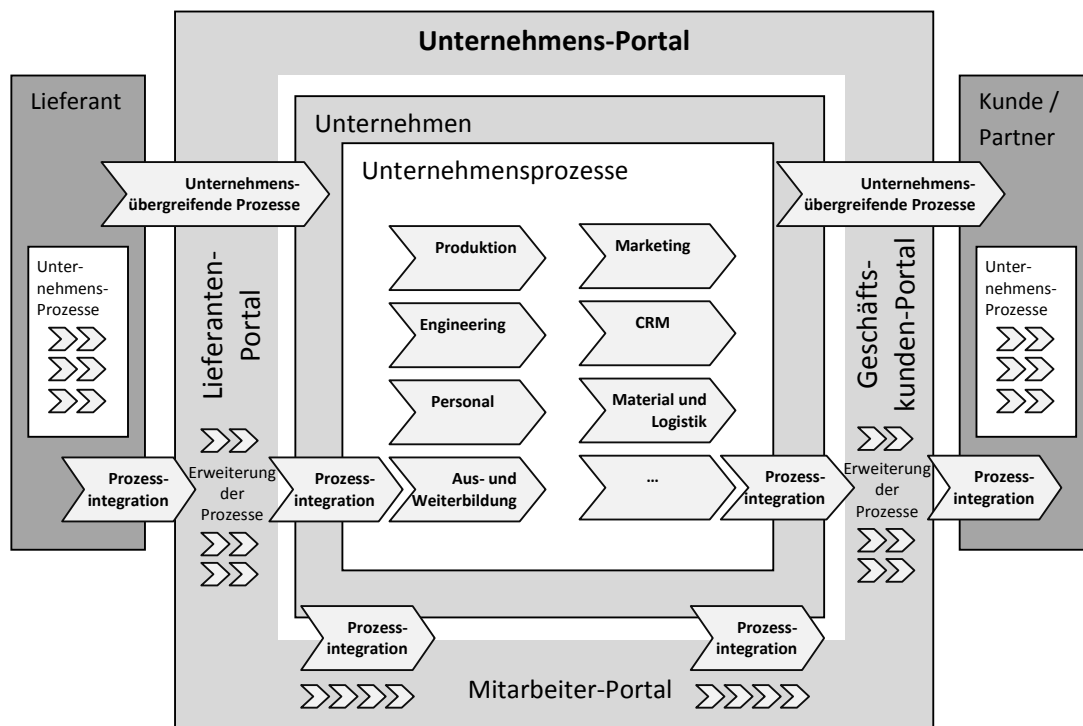


Abbildung 30: Unternehmensportale (Quelle: [40])

### 3.6.3. Make-or-Buy

Nachdem oben aufgezeigt wurde, was Portal-Lösungen leisten, stellt sich die Frage, warum man nicht eine fertige Software-Lösung nutzen und daraus ein Alumni-Portal machen sollte. Die Gründe gegen eine Standard-Portal-Lösung werden im Folgenden erläutert:

Die meisten Portal-Lösungen auf dem Markt sind auf Unternehmens-Portale ausgerichtet. Zielsetzung von Unternehmens-Portalen ist die Bereitstellung beziehungsweise die Unterstützung eines durchgängigen Geschäftsprozess innerhalb eines Unternehmens. Unternehmens-Portale legen den Fokus auf die Erschließung von Wertschöpfungsketten

beziehungsweise Produktmärkte und der Verbindung von Kunden, Mitarbeitern, Zulieferern, Dienstleistern, Herstellern und anderen Kooperationspartnern.

Portal-Lösungen bieten ein umfangreiches Portfolio an Funktionalität. Die mit der Portal-Lösung erworbenen Portal-Basisdienste müssen nicht mehr implementiert werden. Man kann sich auf die Entwicklung der eigenen, individuellen Portal-Anwendungen konzentrieren. Doch je nach Anwendungsfall benötigt man unter Umständen nicht alle Basisdienste einer Portal-Lösung. Diese würden dann nicht genutzt werden.

Der von den Portal-Lösungen realisierte Funktionsumfang, in Form von Portal-Basisdiensten, erspart zwar viel Entwicklungsarbeit, doch die eigenen Portal-Anwendungen müssen noch in das System integriert werden. Portal-Lösungen stellen nur den Rahmen bereit, indem die eigene applikationsspezifische Funktionalität implementiert werden muss. Aus diesem Grund ist trotz einer „fertigen“ Portal-Lösung noch zusätzliche Entwicklungsarbeit notwendig.

Portal-Anwendungen werden über eine Softwareschnittstelle vom Portal aufgerufen. Die sogenannte Portal-API bildet die Grundlage für die Entwicklung von Portal-Anwendungen. Damit eigene Anwendungen in das Portal integriert werden können, muss sich mit den hersteller- und produktspezifischen Schnittstellen auseinandergesetzt werden. Sie erfordern Einarbeitungszeit, denn einheitliche Standards<sup>1</sup> entwickeln sich erst langsam.

Bei vordefinierten Software-Lösungen besteht immer das Problem, dass man Einschränkungen hinsichtlich der Funktionalitäten in Kauf nehmen muss. Verwendet man ein fertiges System, so steht man vor der Aufgabe, die eigenen Prozesse auf das System abzubilden. Irgendwann tauchen Probleme auf, die nicht oder nur mit sehr viel Aufwand zu lösen sind. Oft zwingt das System dazu, einen Weg einzuschlagen, der so ganz gegen die anfängliche Spezifikation geht, nur um die gewünschte Funktionalität zu realisieren. Es funktioniert zwar in den meisten Fällen, doch oft wirft dies wieder neue Probleme auf.

Da Portal-Lösungen komplexe und mächtige Software-Anwendungen sind, sollte man die nicht unwesentlichen Kosten eines solchen Systems berücksichtigen. Vor allem, wenn man nicht alle Portal-Basisdienste in Anspruch nehmen kann oder will. Nach der Markteinteilung nach Gartner [33] ist IBM mit *WebSphere-Portal* Marktführer im Segment der Portal-Lösungen. *WebSphere-Portal* bietet aufgrund seiner offenen Architektur, dem Konzept der Service-orientierten Architektur und der Web 2.0 Technologien, seinem

---

<sup>1</sup> Portlet-API (JSR-168) und Web- Services-for-Remote-Portlets (WSRP)



Nutzer die Möglichkeit, über 5000 Portlets und Dienste aus IBMs-Portal-Lösungskatalog zu integrieren. Doch das spiegelt sich auch im Preis wieder. Der IBM *WebSphere-Portal-Server* kostet 51.500 US \$<sup>1</sup> pro Prozessor. Ein flexibles Lizenzmodell wird ebenfalls angeboten. Für 20 Nutzer-Lizenzen und 12 Monate Support werden 2.580 US \$<sup>2</sup> verlangt.

Nachdem nun erarbeitet wurde, was gegen eine Portal-Software-Lösung spricht, stellt sich nun die Frage nach den Alternativen. Es wurde gezeigt, dass fertige Software-Systeme für Portale nicht universell eingesetzt werden können. Ein Grund ist die Spezialisierung auf Unternehmens-Portale. Viele der am Markt verfügbaren Portal-Software-Lösungen sind Unternehmens-Portal-Lösungen. Weiterhin benötigt man nicht immer alle Basisdienste, welche eine Portal-Software-Lösung mitbringt. Die heterogene Anwendungslandschaft macht es zudem notwendig, die fertigen Systeme so allgemein wie möglich zu entwerfen, damit das Einsatzgebiet weit gefächert werden kann. Umso mehr Entwicklungsarbeit ist für die Integration der eigenen Funktionalität erforderlich. Dadurch kommt man oft an einen Punkt, an dem man über eine eigene Lösung nachdenkt.

### 3. 6. 4. Die eigene Portal-Lösung

Die Entwicklung eines Software-Systems von Grund auf, hat sowohl Vor- als auch Nachteile. Man ist keinen Restriktionen unterlegen und hat alle Freiheiten. Das System kann so umgesetzt werden, dass es den eigenen Vorstellungen entspricht. Somit kann es die vorher spezifizierten Funktionalitäten erfüllen, ohne Einschränkungen in Kauf zu nehmen.

Doch eine Neuentwicklung birgt auch Gefahren. Die Erfahrungen, die die Portal-Software-Hersteller in den Jahren gesammelt haben, machen ihr Produkt zuverlässig und erfolgreich. Fehlende Erfahrung kann sich sowohl in der Implementierung als auch in der konzeptionellen Ebene bemerkbar machen und kostet unnötig Zeit und Geld.

---

<sup>1</sup> Quelle: [http://www-05.ibm.com/de/entwicklung/ueberuns/aktuelles/2007\\_04\\_19.html](http://www-05.ibm.com/de/entwicklung/ueberuns/aktuelles/2007_04_19.html) [Stand: 19.04.2007]

<sup>2</sup> Quelle: <http://www-306.ibm.com/software> [Stand: 26.07.2007]

### 3.6.5. Alumni-Portal-Lösung

Nach Abwegen des Für und Wider der eigenen Entwicklung eines Portals, ist eine Entscheidung für die eigene Portal-Lösung getroffen worden. Die Gründe dafür sind die Folgenden:

Eine Analyse der benötigten Funktionalitäten des zukünftigen Alumni-Portals führte zu dem Schluss, nicht alle Portal-Basisdienste einer Portal-Lösung zu benötigen. Zwar unterscheiden sich die unterschiedlichen Lösungen der Hersteller, indem einige mehr, die anderen weniger Funktionen mit sich bringen. Doch da die meisten Portal-Lösungen auf Unternehmens-Portale fokussiert sind, enthalten diese oft unternehmensportal-spezifische Funktionen. Diese Dienste benötigt das zukünftige Alumni-Portal nicht und sie würden vermutlich auch nicht genutzt werden.

Zudem steht bei Unternehmens-Portalen die Prozessorientierung im Mittelpunkt. Die Dienste in einem Unternehmens-Portal sind zu einer Wertschöpfungskette verknüpft, um bestimmte Geschäftsprozesslogik abzubilden. Das Alumni-Portal hat eher die Vernetzung ehemaliger Kommilitonen zu einer großen Gemeinschaft als die Abbildung von Geschäftsprozessen zum Ziel. Zwar bietet das neue Absolventen-Portale auch Informationen und Dienste an, doch meist sind diese unabhängig und nicht in einer Prozesskette verknüpft. Im Gegensatz zu Unternehmens-Portale hat der Wegfall eines Dienstes keinen Einfluss auf die Funktion des Portals als Ganzes. Dienste im Alumni-Portal können als *Nice-to-have* angesehen werden, demgegenüber steht und fällt ein Unternehmens-Portal mit seinen Diensten.

Ein weiterer Punkt gegen eine Portal-Lösung vom Markt ist der begrenzte finanzielle Rahmen. Die Kosten einer fertigen Portal-Software hätten das Budget des Projektes fast gänzlich aufgebraucht, ohne die Realisierung eigener Funktionalität. Eine Portal-Lösung liefert lediglich den Rahmen für ein Portal, indessen die eigenen Portal-Anwendungen integriert werden. Die Entwicklung eigener Funktionalität wäre damit von vorn herein begrenzt gewesen.

Eine eigenständige Neuentwicklung benötigt zwar Zeit, doch Einarbeitungszeit in eine Portal-Lösung und die Entwicklung eigener Portal-Anwendungen wäre ohnehin notwendig gewesen. Zwar übersteigt die Entwicklungszeit einer eigenen Lösung die Einarbeitungszeit in eine Standard-Lösung, doch eine eigene Portal-Lösung enthält nur die erforderliche

Funktionalität. Nicht benötigte Dienste, die bei Standard-Lösungen brach liegen würden, werden einfach weg gelassen. Das Ergebnis ist eine perfekt auf den Anwendungsfall zugeschnittene Lösung. Die eigene Lösung des Alumni-Portals enthält das Maximum an Individualität. Man ist keinen Restriktionen unterlegen und benutzt das System so, wie es konzipiert wurde.

## 4. Analyse der Dienste des Alumni-Portals

Ein Portal ist ein zentraler und persönlicher Einstieg in die Informationswelt des Internets oder Intranet, von dem aus Verbindungen zu den relevanten Informationen und Diensten hergestellt werden können. Die Relevanz von Informationen und Diensten in einem Portal ist schwer messbar, stellt aber das wichtigste Kriterium für die Informationsrecherche dar.

Nach der Erarbeitung der Klassifizierungskriterien beschäftigt sich das Kapitel mit der Analyse der Dienste des neuen Ilmenauer Alumni-Portals.

### 4. 1. Dienste des Alumni-Portals

Nachfolgend werden die einzelnen Dienste des zukünftigen Ilmenauer Alumni-Portals beschrieben.

#### 4. 1. 1. Lebenslauf

Der Dienst *Lebenslauf* oder Curriculum Vitae (CV) ist einer der wichtigsten Dienste des Alumni-Portals. Dieser Dienst gestattet es dem Alumni, ein Lebenslauf zu pflegen und anderen dadurch die Möglichkeit zu geben, sich über diejenige Person zu informieren.

#### 4. 1. 2. Veröffentlichungen

Der Dienst *Veröffentlichungen* ist ein weiterer bedeutender Dienst des Portals. Durch ihn können Alumni wissenschaftliche Arbeiten auf einfache Art und Weise publizieren, wodurch eine große Wissensdatenbank entstehen soll.

### 4. 1. 3. Ilmedia

Der Dienst *Ilmedia* erweitert den Dienst *Veröffentlichungen* indem er die publizierten wissenschaftlichen Arbeiten transparent an die Bibliothek der Technischen Universität Ilmenau weiterleitet. Diese werden dort geprüft und katalogisiert.

### 4. 1. 4. News & Events

Der Dienst *News & Events* informiert den Alumni über aktuelle Geschehnisse im Umfeld der TU Ilmenau.

### 4. 1. 5. Alumni-News

Es gibt News und Events, die nicht auf den Universitäts-Web-Seiten erscheinen oder mehr das Portal betreffen. Aus diesem Grund gibt es den Dienst *Alumni-News*, der zusätzliche Neuigkeiten enthält.

### 4. 1. 6. Career-Service

Der Dienst *Career-Service* hilft den Absolventen beim Berufseinstieg und klärt Fragen, wie man sich optimal auf den Berufseinstieg vorbereiten kann und was man beachten muss.

### 4. 1. 7. Jobbörse

Der Dienst der *Jobbörse* ist eine Sammlung von Stellenangeboten für Beruf und Praktikum. Sie ist unterteilt in Angebote der Technischen Universität Ilmenau und in externe Stellenausschreibungen.

### 4. 1. 8. Weiterbildung

Der Dienst *Weiterbildung* informiert darüber, welche Möglichkeiten es gibt, sich nach dem Studium weiterzubilden. Hier werden sowohl Angebote der TU Ilmenau als auch von

anderen Hochschulen in Thüringen bereit gestellt. Dazu zählen Aufbau- und Masterstudiengänge und diverse Praktika.

#### **4. 1. 9. E-Learning**

Der Dienst *E-Learning* bietet den Alumni elektronische beziehungsweise netzgestützte Lernangebote an. Unter E-Learning werden alle Formen von Lernen verstanden, bei denen digitale Medien für die Präsentation und Distribution von Lernmaterialien oder zur Unterstützung zwischenmenschlicher Kommunikation zum Einsatz kommen. Der Dienst E-Learning nutzt zum einen das Bildungsportal Thüringen, um sich über E-Learning Angebote zu informieren und zum anderen bietet er selbst E-Learning in Form von eigenen Kursen an.

#### **4. 1. 10. Bildungsportal Thüringen**

Der Dienst des *Bildungsportals Thüringen* informiert über wissenschaftliche Weiterbildungsangebote aller Thüringer Hochschulen, sowie Initiativen und Projekte im Bereich des E-Learning. Damit erweitert dieser Dienst die Dienste *Weiterbildung* und *E-Learning*.

#### **4. 1. 11. Präsentation der TU Ilmenau**

Dieser Dienst soll die Technische Universität Ilmenau präsentieren. Unter diesem Punkt sollen Informationen rund um die TU Ilmenau erscheinen. Der Dienst wird zusätzlich in den Sprachen Englisch, Arabisch und Französisch angeboten. Damit fördert er das Image der Universität sowohl im In- als auch im Ausland.

#### **4. 1. 12. Administrator-Funktionalität**

Der Dienst der *Administrator-Funktionalität* steht alleine den Administratoren zur Verfügung und unterstützt sie in ihren Aufgabenbereichen. Diese reichen von redaktionellen Aufgaben in Form der Prüfung von Lebensläufen und Publikationen, über Nutzer-Benachrichtigungen bis hin zu der Archivierung von News. Die Administratoren nehmen eine zentrale Stellung in dem Portal ein, denn alle im Portal zur Verfügung

gestellten Informationen, abgesehen von den eingebundenen Informationen der Universität, müssen von ihnen geprüft werden. Damit haben sie großen Einfluss auf die Qualität der publizierten Informationen.

#### **4. 1. 13. Nutzerkonto**

Im Gegensatz zu dem Dienst der *Administrator-Funktionalität* steht der Dienst des *Nutzerkontos* nur den Nutzern des Portals zur Verfügung. Das Nutzerkonto ist ein zentraler Bereich der Alumni. Jeder einzelne Alumni hat ein eigenes, personalisiertes Nutzerkonto, indem er die gemachten Eingaben bezüglich der Registrierung, der veröffentlichten Lebensläufe und Publikationen überprüfen, ändern und löschen kann.

#### **4. 1. 14. AlumniDB**

Die TU Ilmenau führt eine Datenbank aller Studenten beziehungsweise ehemaliger Studenten in Ilmenau. Das Alumni-Portal nutzt die Alumni-Datenbank der TU Ilmenau, um die Nutzer bei der Registrierung im Alumni-Portal als Studenten der Technischen Universität zu verifizieren. Durch den sogenannten *AlumniDB*-Dienst kann eindeutig geprüft werden, ob ein Nutzer des Portals ehemaliger TU Ilmenau Student war oder nicht.

## 4. 2. Klassifikationskriterien

Damit die oben beschriebenen Dienste analysiert werden können, werden im Folgenden Klassifizierungskriterien erarbeitet.

### 4. 2. 1. Integrationslokation

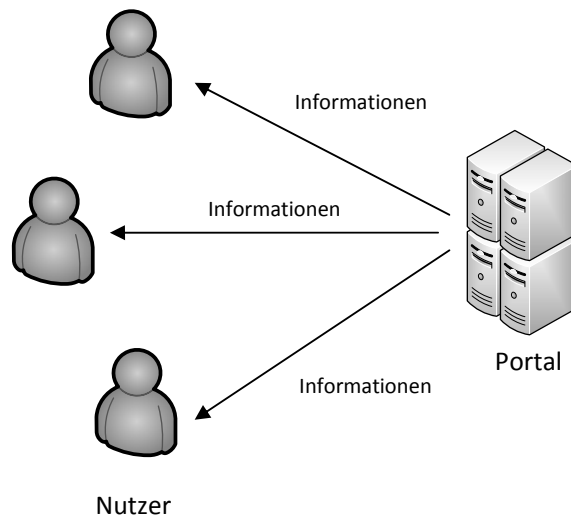
Das Klassifikationskriterium Integrationslokation versucht die Frage, an welcher Stelle im Portal der Dienst genutzt wird, zu beantworten. Dienste in Portalen können sowohl im *Frontend* als auch im *Backend* integriert werden. Die Begriffe Frontend und Backend werden in der Informatik an verschiedenen Stellen in Verbindung mit einer Schichteneinteilung verwendet. Dabei ist typischerweise das Frontend näher am Benutzer, das Backend näher am System.

Ist ein Dienst dem Frontend zuzuordnen, so nimmt der Nutzer den Dienst direkt in Anspruch. Er nutzt den Dienst, indem er die Anfragen selbst formuliert und die Antworten direkt zurück bekommt. Demgegenüber ist ein Dienst im Backend dem Nutzer verborgen. Er nimmt den Dienst in Anspruch, doch meist unmerklich. Der Dienst wird nicht vom Nutzer bedient, sondern von der im Hintergrund arbeitenden Applikation.

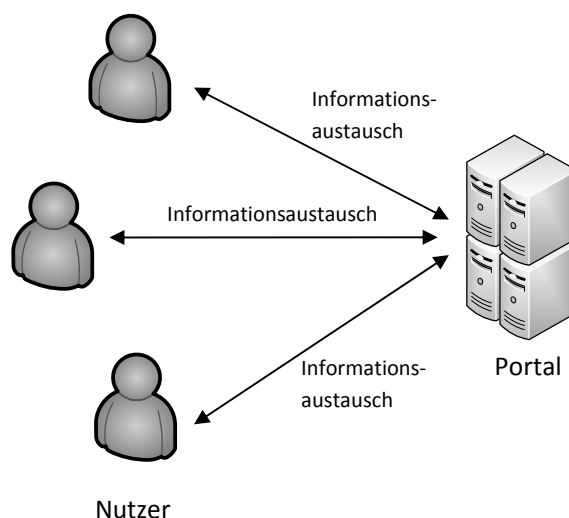
### 4. 2. 2. Informationsfluss

Wie in Kapitel 2.2.4 erläutert wurde, sind Portale Web 2.0 Anwendungen. Im Zentrum des Web 2.0 stehen die Daten der Nutzer, welche in einer Datenbank der Web 2.0 Anwendung gespeichert werden. Die Web 2.0 Dienste stellen für die Daten der Nutzer den Rahmen beziehungsweise die erforderliche Infrastruktur bereit. Die Nutzer gestalten mit ihren Daten die Web 2.0 Dienste erheblich mit. Die von den Nutzern erzeugten Daten sind das „Futter“ dieser Dienste. Sie werden in den Diensten gesammelt und gebündelt. Je mehr nutzergenerierte Daten ein Web 2.0 Dienst zur Verfügung hat, desto mehr Informationen lassen sich dort finden und umso attraktiver wird der Dienst.



**Abbildung 31: Informationsfluss im Web 1.0**

Ist das Web 1.0 gekennzeichnet durch einen *unidirektionalen Informationsfluss* vom Dienstanbieter zum Dienstanutzer, so rückt beim Web 2.0 der Nutzer in den Vordergrund. Er ist nicht nur Konsument, sondern auch Produzent der Informationen. Die Dienste einer Web 2.0 Anwendung sind auf die Beteiligung der Nutzer ausgerichtet, die ihre Informationen gegenseitig austauschen. Jeder Anwender kann seine Daten und Ideen auf dieser Plattform bereitstellen und veröffentlichen. Durch die Vernetzung der einzelnen Nutzer und Daten ergibt sich dadurch ein Kollektiv, das eine eigene Intelligenz besitzt. Web 2.0 Anwendungen sind gekennzeichnet durch einen *bidirektionalen Informationsfluss*.

**Abbildung 32: Informationsfluss im Web 2.0**

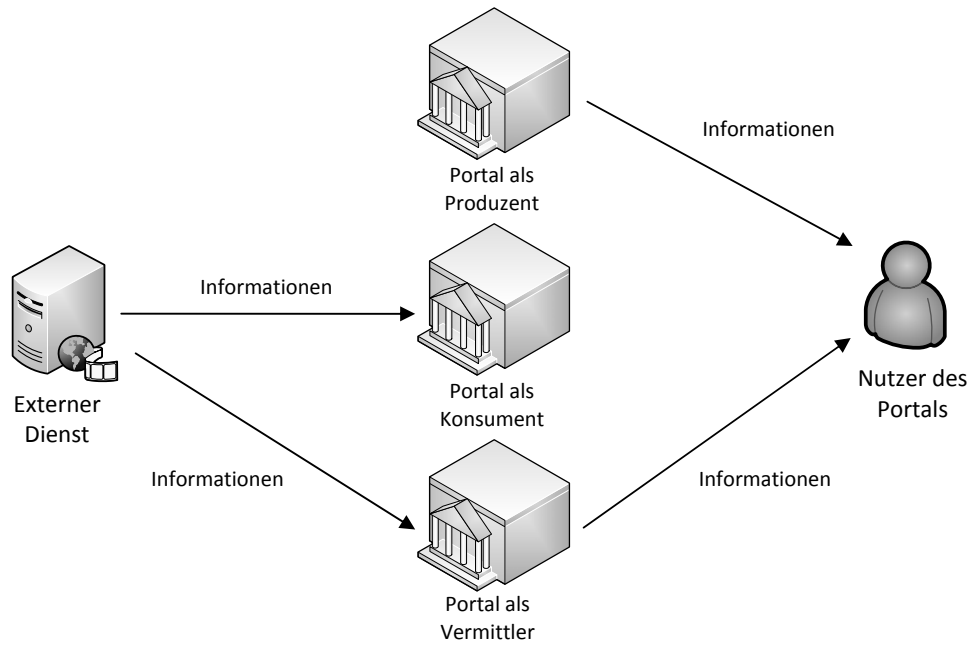
### 4. 2. 3. Kommunikationsmodell

Um Dienste in Anspruch zu nehmen, bedarf es der Kommunikation. Kommunikation bezeichnet die Vermittlung, den Austausch und die Aufnahme von Informationen. In Client-Server-Architekturen unterscheidet man den Dienstanutzer und den Dienstanbieter. Der Dienstanutzer nimmt Dienste in Anspruch, der Dienstanbieter hingegen bietet welche an. Damit der Dienstanutzer einen Dienst nutzen kann, muss er mit dem Dienstanbieter kommunizieren. Dabei unterscheidet man zwischen *synchroner* und *asynchroner Kommunikation* (vgl. Kapitel 2.3.3).

### 4. 2. 4. Dienstersprung

Ein Portal bezeichnet einen zentralen Zugang, über den man auf individuell zugeschnittene Informationen und Dienste zugreifen kann. Die angebotenen Dienste müssen nicht von dem Portal selbst realisiert sein. Aus diesem Grund ist die Integration ein wichtiger Aspekt innerhalb von Portalen (vgl. Kapitel 2.1.6). Somit kann ein Portal ein umfangreiches Portfolio an Diensten besitzen, die es durch Integration eingebunden hat.

Hat ein Portal einen eigenen Dienst realisiert, so ist das Portal *Produzent* des Dienstes. Demgegenüber stehen die Möglichkeiten einen Dienst zu integrieren und diesen selbst zu konsumieren oder ihn dem Nutzer transparent zu präsentieren. Ist ein Portal *Konsument* eines Dienstes, so wird der integrierte Dienst für die Realisierung anderer Dienste verwendet. Meist werden verschiedene Dienste integriert und zu einem anderen Dienst aggregiert, der dann dem Nutzer präsentiert wird. Als letzte Möglichkeit kann das Portal als *Vermittler* tätig sein, indem Dienste in das Portal integriert und diese dem Nutzer präsentiert werden. Dabei werden keinerlei Änderungen am Layout und an der Funktionsweise des integrierten Dienstes vorgenommen.

**Abbildung 33: Dienstursprung**

## 4.3. Analyse

### 4.3.1. Überblick

<i>Dienst</i>	<i>Integrationslokation</i>	<i>Informationsfluss</i>	<i>Kommunikationsmodell</i>	<i>Dienstursprung</i>
<b>Lebenslauf</b>	Frontend	Nutzer ⇌ Dienst (bidirektional)	synchrones oder blockierendes Kommunikationsmodell	Produzent
<b>Veröffentlichungen</b>	Frontend	Nutzer ⇌ Dienst (bidirektional)	synchrones oder blockierendes Kommunikationsmodell	Produzent
<b>ilmedia</b>	Backend	Dienst ⇒ Dienst (unidirektional)	asynchrones oder nicht-blockierendes Kommunikationsmodell	Konsument
<b>News &amp; Events</b>	Frontend	Nutzer ⇌ Dienst (unidirektional)	synchrones oder blockierendes Kommunikationsmodell	Vermittler
<b>Alumni-News</b>	Frontend	Nutzer ⇌ Dienst (unidirektional)	synchrones oder blockierendes Kommunikationsmodell	Produzent

<b>Career-Service</b>	Frontend	Nutzer ⇌ Dienst (unidirektional)	synchrones oder blockierendes Kommunikationsmodell	Vermittler
<b>Jobbörse</b>	Frontend	Nutzer ⇌ Dienst (unidirektional)	synchrones oder blockierendes Kommunikationsmodell	Vermittler
<b>Weiterbildung</b>	Frontend	Nutzer ⇌ Dienst (unidirektional)	synchrones oder blockierendes Kommunikationsmodell	Vermittler
<b>E-Learning</b>	Frontend	Nutzer ⇌ Dienst (unidirektional)	synchrones oder blockierendes Kommunikationsmodell	Vermittler / Produzent
<b>BPT</b>	Backend	Dienst ⇌⇌ Dienst (bidirektional)	synchrones oder blockierendes Kommunikationsmodell	Konsument
<b>Präsentation der TLL-Menau</b>	Frontend	Nutzer ⇌ Dienst (unidirektional)	synchrones oder blockierendes Kommunikationsmodell	Produzent
<b>Administrator- Funktionalität</b>	Frontend	Nutzer ⇌⇌ Dienst (bidirektional)	synchrones oder blockierendes Kommunikationsmodell	Produzent
<b>Nutzerkonto</b>	Frontend	Nutzer ⇌⇌ Dienst (bidirektional)	synchrones oder blockierendes Kommunikationsmodell	Produzent

AlumniDB	Backend	Dienst ⇄ Dienst (bidirektional)	synchrones oder blockierendes Kommunikationsmodell	Konsument
----------	---------	---------------------------------------	--	-----------

### 4.3.2. Analyse im Detail

Für eine detaillierte Analyse der Dienste des zukünftigen Alumni-Portals sei auf den Anhang A verwiesen.

## 4.4. Zusammenfassung

Dieses Kapitel analysierte die Dienste des zukünftigen Alumni-Portals der TU Ilmenau nach vorher erarbeiteten Kriterien. Dabei wurde zuerst zwischen Diensten im *Backend* und Diensten im *Frontend* unterschieden. Ist ein Dienst dem Frontend zuzuordnen, so nimmt der Nutzer den Dienst direkt in Anspruch. Demgegenüber ist ein Dienst im Backend dem Nutzer verborgen. Der Dienst wird nicht vom Nutzer bedient, sondern von dem im Hintergrund arbeitenden System. Weiterhin wurde analysiert, in welche Richtungen die Informationen des Dienstes fließen. Daraus kann abgeleitet werden, ob dieser Dienst dem Web 1.0 oder dem Web 2.0 folgt. Charakteristisch für Web 2.0 Anwendungen ist ein *bidirektionaler Informationsfluss*. Außerdem wurde die Kommunikation mit dem Dienst betrachtet. Es wurde zwischen einer *synchronen* und einer *asynchronen Kommunikation* unterschieden. Da die meisten Dienste des Portals im Frontend angesiedelt sind, erzwingt dies eine synchrone Kommunikation mit dem jeweiligen Dienst. Als letztes Kriterium wurde das Portal als *Konsument*, *Produzent* oder *Vermittler* des Dienstes beurteilt. Das Portal tritt in annähernd gleich vielen Fällen als Produzent oder Vermittler von Diensten auf. Dies spiegelt den Integrationsgedanken von Portalen wieder. Das Portal ist dabei der zentrale Zugang zu Informationen und Diensten, unabhängig von deren Lokation.





## 5. Konzeption der Schnittstellen

Portale können nur dann erfolgreich sein, wenn sie den Nutzern eine Fülle an Informationen und Diensten bereitstellen. Ein charakteristisches Merkmal eines Portals ist die Integration. Dem Portal muss eine Infrastruktur zu Grunde liegen, die es ermöglicht, andere Dienste einzubinden. Dabei darf es keine Rolle spielen, ob diese Anwendungen an unterschiedlichen Standorten betrieben werden. Ein Portal spannt ein Dach über diese Anwendungssysteme und stellt deren Funktionen und Daten in einer gemeinsamen Benutzeroberfläche zur Verfügung. Notwendige Voraussetzung einer Integration beziehungsweise Interaktion ist eine gemeinsame Schnittstelle. Eine Schnittstelle definiert im Allgemeinen eine Verbindungsstelle zwischen zwei miteinander in Beziehung stehenden Systemen. [34]

Das zukünftige Alumni-Portal soll Schnittstellen zu der Bibliothek der TU Ilmenau, dem Bildungsportal Thüringen und der Alumni-Datenbank der TU Ilmenau aufweisen. Durch diese Schnittstellen können die Dienste des jeweiligen Systems in das Alumni-Portal integriert werden. In diesem Kapitel werden die Dienste analysiert und aufbauend auf den Ergebnissen, die Schnittstellen konzipiert.

## **5. 1. Bildungsportal Thüringen**

### **5. 1. 1. Allgemeines zum BPT-Dienst**

Das *Bildungsportal Thüringen* (BPT) ist das Forum für wissenschaftliche Weiterbildung in Thüringen. Es bündelt Informationen über wissenschaftliche Weiterbildungsangebote aller Thüringer Hochschulen, sowie Initiativen und Projekte im Bereich des E-Learning.

Das Portal ist ein Gemeinschaftsprojekt der Universitäten Jena, Ilmenau, Weimar und Erfurt, der Hochschule für Musik in Weimar sowie der Fachhochschulen Jena, Erfurt, Schmalkalden und Nordhausen. Es wird vom Thüringer Ministerium für Wissenschaft, Forschung und Kunst gefördert.

Sowohl bildungsinteressierte private Nutzer als auch Institutionen und Unternehmen finden seit dem 5.6.2002 im BPT einen tagesaktuellen Überblick über qualitativ hochwertige Weiterbildungsangebote der thüringischen Hochschulen. Es können sowohl klassische, als auch digitale Weiterbildungsangebote beziehungsweise digital aufbereitete Lehrinhalte und Projekte in das Portal eingestellt werden. Das Bildungsportal Thüringen nimmt zwischen den Anbietern und den Nutzern der Aus- und Weiterbildungsangebote eine vermittelnde Position ein. Im Portal werden ausschließlich die Metadaten der Angebote dargestellt, hingegen die Inhalte beim Anbieter verbleiben. Dies gilt ebenfalls für die Rechte an den Inhalten. Die Anbieter sind zur Erhaltung von Qualität und Aktualität ihrer Inhalte und Metadaten selbst verantwortlich.

### **5. 1. 2. Nutzen des BPT-Dienstes**

Das Alumni-Portal will das Bildungsportal Thüringen als Dienst nutzen, um sich über Weiterbildungsangebote, sowie Initiativen und Projekte im Bereich des E-Learning der thüringischen Hochschulen zu informieren und diese den Nutzern des Portals zu präsentieren.

### 5.1.3. Analyse der BPT-Schnittstelle

Das Bildungsportal Thüringen bietet bereits eine Schnittstelle an. Diese ist für die Partner des Bildungsportals gedacht, damit diese Informationen aus dem Portal exportieren können. Nachfolgende Abbildung verdeutlicht die Interaktion.

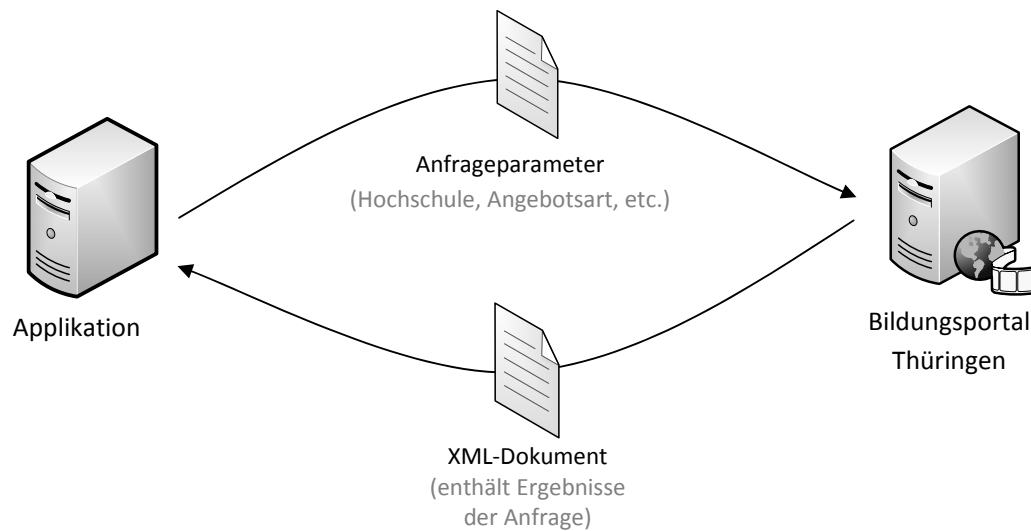


Abbildung 34: Schnittstelle zum Bildungsportal Thüringen

Da bereits eine Schnittstelle zum Bildungsportal existiert, wird im folgenden die Umsetzung analysiert, um zu prüfen, ob diese Schnittstelle für die Interaktion mit dem Alumni-Portal verwendet werden kann.

Die Schnittstelle zum Bildungsportal Thüringen ist durch ein server-seitiges Script (vgl. Kapitel 2.2.2) realisiert, das die Ergebnisse einer Anfrage per XML (vgl. Kapitel 2.2.3) zurückliefert. Die Anfragen an das Bildungsportal werden durch verschiedene Parameter formuliert, die dem Servlet übermittelt werden. Der erste Parameter ist `hochschule`. Durch ihn kann die Suche bezüglich einzelner Hochschulen eingegrenzt werden. Damit ist es möglich, gezielt Weiterbildungsmöglichkeiten oder E-Learning-Angebote an einer bestimmten Hochschule zu suchen.

Folgende Eingaben des Parameters `hochschule` sind erlaubt:

- Friedrich-Schiller-Universität Jena
- Bauhaus-Universität Weimar
- Fachhochschule Jena
- Technische Universität Ilmenau
- Fachhochschule Erfurt
- Universität Erfurt
- Fachhochschule Nordhausen
- Fachhochschule Schmalkalden
- Hochschule für Musik FRANZ LISZT Weimar
- Berufsakademie Gera//Eisenach
- Sonstige Kooperationspartner

Der zweite Parameter ist `angebotsart`. Durch ihn werden die angeforderten Informationen weiter konkretisiert. So ist es möglich nur Weiterbildungsangebote anzufordern und die restlichen Angebotsarten werden bei der Suche nicht beachtet. Folgende Eingaben sind möglich:

- Weiterbildungsangebot
- Projekt
- Neue Medien/E-Learning
- Veranstaltung/Veranstaltungsreihe

Diese beiden Parameter genügen noch nicht, um eine Suche über diese Schnittstelle zu initiieren. Da die Schnittstelle nur für Partner des Bildungsportals gedacht ist, sind zwei weitere Parameter notwendig, die den jeweiligen Partner authentisieren. Das ist zum einen die registrierte Email-Adresse und zum anderen das zugehörige Passwort. Erst mit diesen zusätzlichen Daten kann eine Suche erfolgen.

Die Partner des Bildungsportals erhalten nach einer initiierten Suche die Antwort in Form eines XML-Dokumentes. Dieses Dokument enthält die Anfragergebnisse für die individuelle Auswertung. Nachfolgendes Listing zeigt ein solches XML-Dokument.

```

<?xml version="1.0" encoding="UTF-8" ?>
<courses>
  <course>
    <NAME>Web Services</NAME>
    <Kurzbeschreibung>
      Dauer: 60 Stunden, davon ca. 30 Stunden Selbststudium,
      Das Modul soll die grundlegende Idee und die Technologie
      vorstellen.
    </Kurzbeschreibung>
    <Hochschule>Technische Universität Ilmenau</Hochschule>
    <Kooperationspartner />
    <Verantwortlicher>Jochen Seitz</Verantwortlicher>
    <EMail_Verantwortlicher>
      jochen.seitz@tu-ilmenau.de
    </EMail_Verantwortlicher>
    <Master>0</Master>
    <Taetigkeitsfeld_Zielgruppe>
      Dozenten, Berufstätige, Berufliche Neuorientierung,
      Berufsrückkehrer
    </Taetigkeitsfeld_Zielgruppe>
    <Lebensphase>unabhängig</Lebensphase>
    <Lernziel>
      Wissensvermittlung über Angebote von Internetdiensten,
      SOAP, WSDL, UDDI; Webinhalte erstellen und Pflegen
    </Lernziel>
    <Besonderheiten>
      Berufsbegleitend, Wochenendveranstaltung
    </Besonderheiten>
    <Dauer />
    <Kosten>200 bis 500 Euro</Kosten>
    <Angebotsart>Weiterbildungsangebot</Angebotsart>
    <Schlagwort>
      Web Services, Internetdienste, SOAP, WSDL, UDDI
    </Schlagwort>
    <Link_zum_Angebot>http://www.tkm.de</Link_zum_Angebot>
    <BPT_Granularitaet>Modul/Kurs</BPT_Granularitaet>
    <Titel_Verantw_1>Prof. Dr. rer. nat. habil.
    </Titel_Verantw_1>
    <Vorname_Verantw_1>Jochen</Vorname_Verantw_1>
    <Nachname_Verantw_1>Seitz</Nachname_Verantw_1>
    <Hochschule_Verantw_1>
      Technische Universität Ilmenau
    </Hochschule_Verantw_1>
    <Fakultaet_Verantw_1>
      Fakultät Elektrotechnik und Informationstechnik
    </Fakultaet_Verantw_1>
    <Einrichtung_Verantw_1 />
    <Abschluss>
      Teilnahmebescheinigung mit/ohne Prüfungsnote
    </Abschluss>
    <Studienform>Präsenzstudium, Teilzeitstudium</Studienform>
    <Studienart>Weiterbildung</Studienart>
    <NM_frei_zugaenglich>0</NM_frei_zugaenglich>
    <NM_registrierungspflichtig>0</NM_registrierungspflichtig>
    <NM_Dienste_Systeme>0</NM_Dienste_Systeme>
    <NM_Dienste_Betreuung>0</NM_Dienste_Betreuung>
    <NM_Dienste_Entwicklung>0</NM_Dienste_Entwicklung>
    <NM_Projekte>0</NM_Projekte>
    <WB_Projekt>0</WB_Projekt>
  </course>
</courses>

```

```

<BPT_Branchen>übergreifend, IuK</BPT_Branchen>
<BPT_Fachgebiete>Ingenieurwissenschaften</BPT_Fachgebiete>
<BPT_Status>aktuell</BPT_Status>
<Langbeschreibung>
  Das Modul Informationsmanagement hat einen
  Gesamtumfang von 60 Stunden...
</Langbeschreibung>
<BACKLINK>
  http://www.bildungsportal-thueringen.de:80/s...
</BACKLINK>
<Weiterbildung>Weiterbildungskurse</Weiterbildung>
<Durchfuehrungen>
<Beginn>1.4.2007</Beginn>
</Durchfuehrungen>
<Direktlink>
  http://www.bildungsportal-thueringen.de/tkm-ws/
</Direktlink>
</course>

...

</courses>

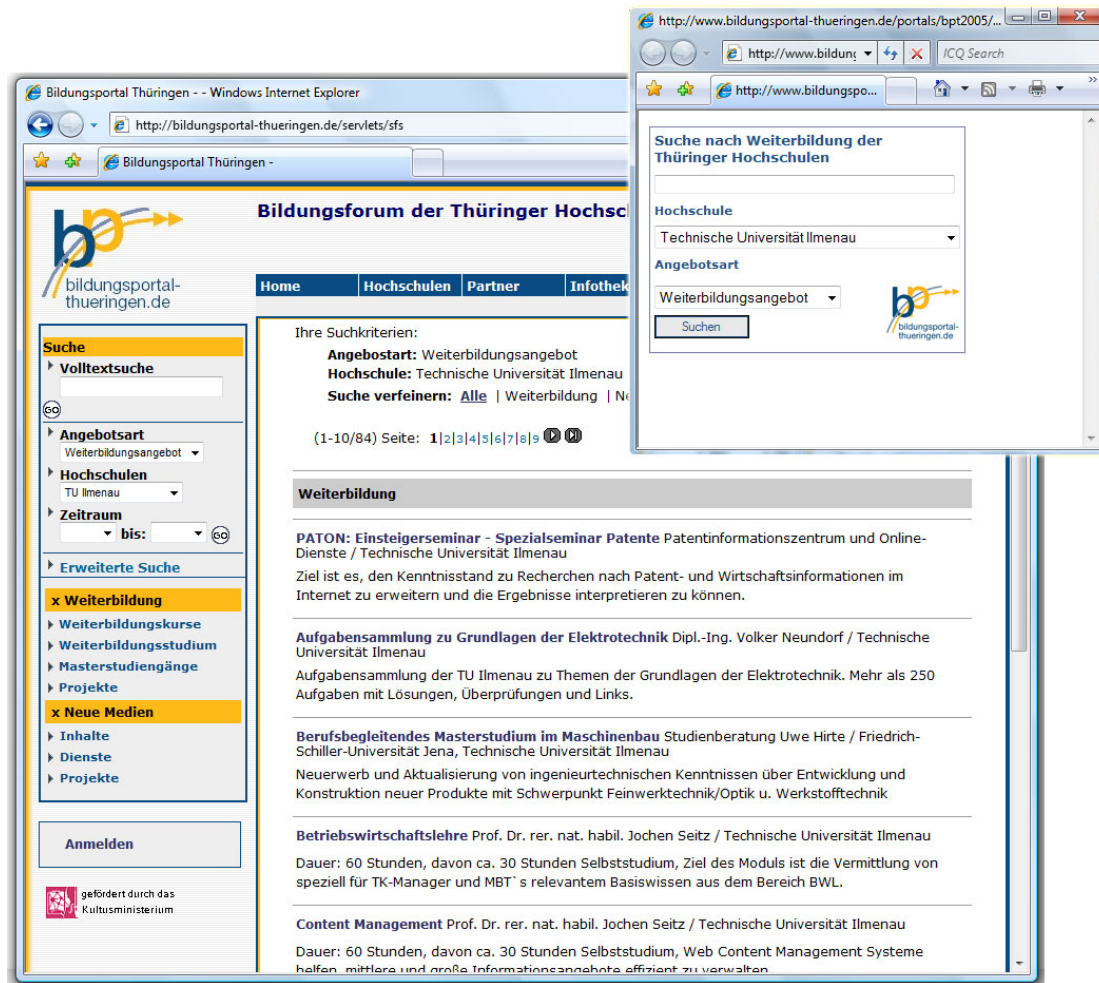
```

Listing 2: XML-Dokument vom BPT

Das obige XML-Dokument ist das Ergebnis der Anfrage mit den Parametern `hochschule=Technische Universität Ilmenau` und `angebortsart=Weiterbildung`. Exemplarisch wurde nur ein Weiterbildungsangebot präsentiert, das vollständige XML-Dokument enthält deutlich mehr Angebote. Jedes Weiterbildungsangebot ist durch mehrere Elemente spezifiziert. Dazu gehören der Name und eine kurze Beschreibung des Angebots, der Veranstaltungsort, die Lernziele und weitere.

Auch den Nicht-Partnern soll die Möglichkeit gegeben werden, in der Datenbank des Bildungsportals nach Informationen zu suchen. Dazu bietet das Portal auf der Web-Seite<sup>1</sup> ein Formular für die Suche an. Die Ergebnisse werden anschließend in vorformatierter Form präsentiert. Zudem ist es möglich diese Suche auch auf der eigenen Web-Seite einzubinden. Dazu bietet das Bildungsportal eine Reihe von vordefinierten Formularen, die mühelos in die eigene Web-Seite integriert werden können. Wird das Formular ausgefüllt und abgeschickt, so wird der Nutzer auf die Bildungsportal-Seite weitergeleitet und ihm das Ergebnis der Suche präsentiert. Die nachfolgende Abbildung verdeutlicht den Vorgang.

<sup>1</sup> Link: <http://www.bildungsportal-thueringen.de>



Die obige Präsentation des Ergebnisses der Suche ist wenig flexibel. Es lassen sich keinerlei Anpassungen bezüglich der Darstellung der Daten vornehmen. Nicht-Partner müssen allein mit dieser Form der Darstellung vorlieb nehmen.

#### 5. 1. 4. Eignung für das Alumni-Portal

Wie im vorherigen Abschnitt erläutert, bietet das Bildungsportal Thüringen bereits eine Schnittstelle an. Nun ist zu prüfen, ob diese Schnittstelle für die Anbindung des Alumni-Portals an das Bildungsportal geeignet ist.

Das Alumni-Portal will den Dienst des Bildungsportals Thüringen nutzen, um sich über Weiterbildungsangebote, sowie Initiativen und Projekte im Bereich des E-Learning der

thüringischen Hochschulen zu informieren. Dafür muss die Schnittstelle die Möglichkeit bieten, Anfragen entgegen zu nehmen und diese auszuwerten. Das Format der zurück gelieferten Ergebnisse der Anfrage sollte gut durch die eigene Anwendung anpassbar sein.

Die vorhandene Schnittstelle des Bildungsportals bietet die Möglichkeit Anfragen zu formulieren. Das Ergebnis der Anfrage wird in Form eines XML-Dokument zurückgeliefert. Mit diesem XML-Dokument kann anschließend unterschiedlich verfahren werden.

Zum einen ist es möglich die Informationen des Dokumentes weiter zu verarbeiten, indem das XML-Dokument geparkt wird und die notwendigen Informationen ausgelesen werden. Zum anderen ist es möglich das XML-Dokument mittels eines XSLT-Stylesheets in eine benutzerfreundliche Darstellung in Form von HTML, DOC, PDF oder ähnliches zu transformieren (vgl. Kapitel 2.2.3).

Dank den umfangreichen Anwendungsmöglichkeiten von XML, was zweifellos zu der Popularität und Akzeptanz dieses Datenformats führte, kann XML als universelles Austauschformat eingesetzt werden. Hat man Daten als XML-Dokument vorliegen, so gibt es fast unbegrenzte Möglichkeiten diese Informationen zu verarbeiten.

Aus dem Grund ist diese XML-Schnittstelle auch für das Alumni-Portal hervorragend anwendbar. Das von der Schnittstelle gelieferte XML-Dokument bietet genug Freiheiten, die Informationen für den eigenen Anwendungsfall anzupassen. Die Konzeption einer neuen Schnittstelle ist nicht notwendig.



## 5.2. Ilmedia

### 5.2.1. Allgemeines zum Ilmedia-Dienst

Der Dienst Ilmedia der Universitätsbibliothek der TU Ilmenau bündelt in den drei Dienstleistungsbereichen Bücher veröffentlichen, Elektronisch Publizieren und Hochschulbibliographie alle Aktivitäten für Publikationen aus der TU Ilmenau, die von deren Angehörigen genutzt werden können.

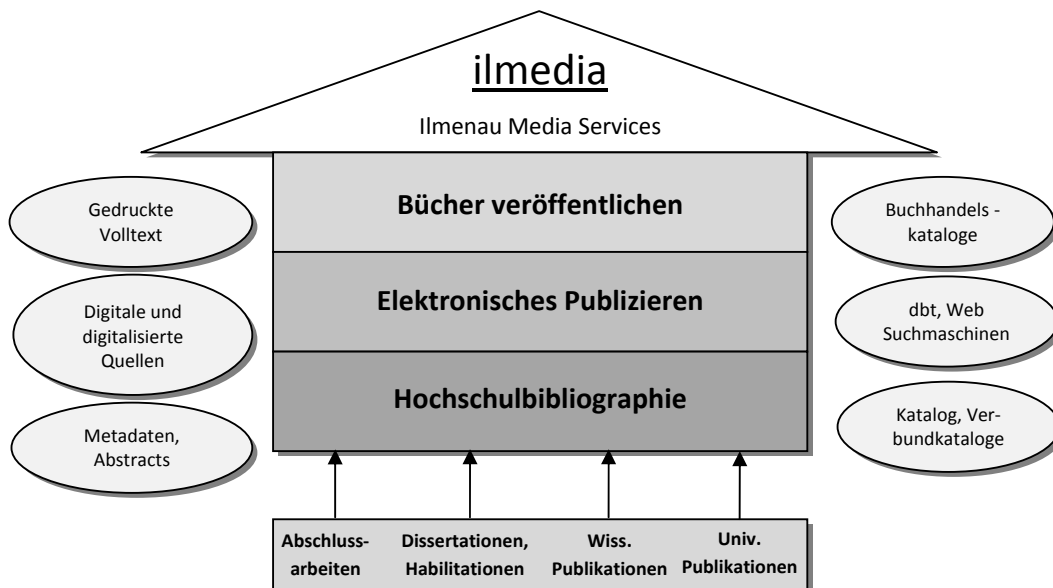


Abbildung 35: Ilmedia (Quelle: [35])

Das Angebotsspektrum reicht von der bloßen Verzeichnung bis hin zum Druck von Büchern. Ziel aller Aktivitäten ist die Förderung der Verbreitung und Rezeption der an der Universität erstellten wissenschaftlichen Arbeiten und damit die Stärkung des wissenschaftlichen Profils der TU Ilmenau. [35]

### **5. 2. 2. Nutzen des Ilmedia-Dienstes**

Hier liegt die Idee zugrunde, den Dienst der Ilmedia zu nutzen, um die im Alumni-Portal registrierten Publikationen an die Bibliothek weiterzuleiten, sodass sie dort veröffentlicht und katalogisiert werden können.

### **5. 2. 3. Konzeption der Ilmedia-Schnittstelle**

Bevor eine konkrete Integrationstechnologie ausgewählt werden kann, muss geklärt werden, wie die zu integrierenden Systeme gekoppelt werden und auf welcher Ebene die Integration ansetzt (vgl. Kapitel 2.3).

Da bei dieser Integration nur zwei Anwendungen miteinander verbunden werden, nämlich das Alumni-Portal und die Bibliothek der TU Ilmenau, ist die geeignetste Integrationstopologie die Punkt-zu-Punkt-Integration. In überschaubaren Integrationsszenarien ist diese Topologie am einfachsten zu realisieren.

Die Katalogisierung und Veröffentlichung einer Publikation in einer Bibliothek ist ein komplexer Vorgang, der möglicherweise Monate dauern kann. Dies ist der Fall, wenn die Publikation schon in irgendeiner anderen Art der Öffentlichkeit zugänglich gemacht wurde. Solche Fälle sind von der Bibliothek ausführlich zu prüfen. Auch wenn es sich um eine erstmalige Veröffentlichung handelt, so durchläuft die Publikation trotzdem gewisse Arbeitsabläufe, die sie Prüfungen verschiedener Art unterziehen. Aus diesen Voraussetzungen lässt sich ableiten, dass eine automatisierte Einspeisung der im Alumni-Portal registrierten Publikationen in den Katalog der Universitätsbibliothek nicht möglich ist. Eine Datenintegration ist deshalb nicht anwendbar. Es bleibt noch die Möglichkeit der Integration auf Präsentations- und Funktionsebene. Eine Integration auf Präsentationsebene bietet Nachteile bei der Performanz sowie eine geringere Flexibilität. Außerdem wird sie nur angewandt, wenn eine Integration auf Daten- und Funktionsebene nicht durchführbar ist.

Wie oben erwähnt unterzieht die Bibliothek der TU Ilmenau jede Publikation einer genauen Prüfung, bevor diese in den Katalog der Bibliothek aufgenommen wird. Ziel der Integration ist es, diese vorhandene Funktionalität wiederzuverwenden beziehungsweise zu nutzen. Deshalb ist für die Bibliothek der TU Ilmenau die Funktionsintegration die

sinnvollste Art der Anwendungsintegration. Nur sie realisiert den Zugriff auf die Logik der zu integrierenden Anwendung.

Die Vielzahl an Integrationstechnologien für die Funktionsintegration macht es notwendig, die Anwendung weiteren Kriterien zu unterziehen (vgl. Kapitel 2.3.3). Die Ergebnisse unterstützen die Entscheidung einer konkreten Integrationstechnologie.

Damit eine Publikation bei Ilmedia katalogisiert werden kann, muss sie bei der Universitätsbibliothek eingereicht werden. Danach beginnt der Arbeitsablauf der Bibliothek, an deren Ende die Katalogisierung beziehungsweise die Veröffentlichung der Publikation steht. Aus diesem Sachverhalt lässt sich ableiten, dass eine asynchrone Kommunikation mit der zu integrierenden Anwendung bevorzugt wird. Diese ist am sinnvollsten, wenn eine Antwort vom Dienstanbieter nicht sofort oder nicht mit voller Wahrscheinlichkeit erwartet werden kann. Es kann nicht sichergestellt werden, dass auf Seiten der Bibliothek immer jemand die Publikation entgegennimmt. Eine Publikations-Warteschlange ist hier eine sinnvolle Variante, um die Verfügbarkeit des Dienstes zu steigern.

Nachdem das Kommunikationsmodell festgelegt ist, stellt sich die Frage nach dem Dienstaufrufmodell. Ein Dienst kann sowohl über eine wohldefinierte Schnittstelle als auch über eine Nachricht aufgerufen werden. Wie man in der Abbildung 27 in Kapitel 2.3.4 sehen kann, ist eine asynchrone Kommunikation eng mit dem nachrichten-basierten und eine synchrone Kommunikation eng mit dem API-basierten Dienstaufrufmodell verknüpft. Ist bereits eine asynchrone Kommunikation festgelegt, so ist meist ein nachrichten-basiertes Dienstaufrufmodell zu bevorzugen.

Als letztes Kriterium bleibt noch der Integrationsgegenstand zu prüfen. Bei der Wahl zwischen der Integration einer Funktion, eines Objekt oder eines Dienstes stellt sich immer die Frage nach einer losen oder festen Kopplung der zu integrierenden Systeme. Da eine lose Kopplung der beteiligten Systeme zu bevorzugen ist, ist eine Entscheidung zwischen dem objektorientierten und dem dienstorientierten Integrationsparadigma zu treffen.

Betrachtet man die oben erarbeiteten Kriterien und die Abbildung 27 in Kapitel 2.3.4, so ist eine Entscheidung zwischen einer MOM und einem Web Service als Integrationstechnologie zu treffen. Für die Ilmedia-Schnittstelle wurde das Konzept der MOM bevorzugt. Zwar integrieren sowohl MOM als auch Web-Service Anwendungssysteme, indem Nachrichten zwischen den beteiligten Systemen ausgetauscht

werden, doch ist das Konzept der MOM ein wenig flexibler hinsichtlich der zu übertragenen Nachrichten. Das Format für die Nachrichten ist nicht festgelegt und der Inhalt kann sowohl aus Daten als auch aus Anweisungen und Kontrollinformationen bestehen.

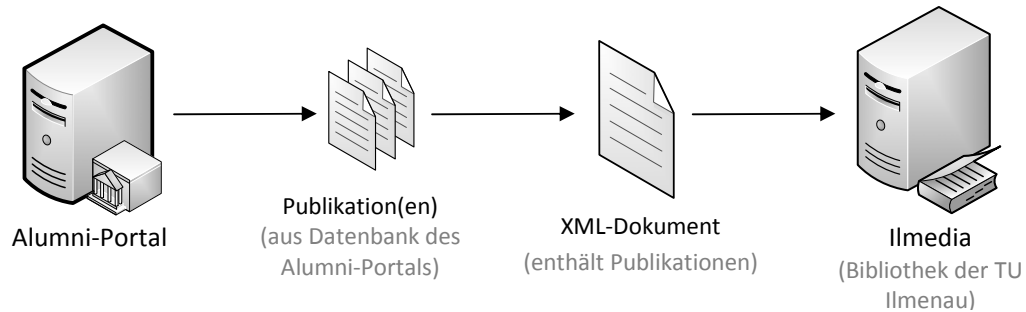
Nachdem die Integrationstechnologie ausgewählt wurde, kann die Ilmedia-Schnittstelle umgesetzt werden.

Das grundsätzliche Konzept einer MOM ist die Realisierung einer Middleware, die durch den Austausch von Nachrichten kommuniziert. Diese Nachrichten sind die aus dem Alumni-Portal exportierten Publikationen, die nun der Bibliothek beziehungsweise der Ilmedia zur Verfügung gestellt werden, damit diese dort geprüft und katalogisiert werden. Welches Format die Daten haben, ist erstmal nicht von Bedeutung.

Eine triviale Möglichkeit ist die Speicherung im Datenformat PDF. Das *Portable Document-Format* (PDF) ist ein von der Firma Adobe entwickeltes Dateiformat, mit dessen Hilfe Dokumente beliebiger Art plattformübergreifend elektronisch veröffentlicht werden können. Trotzdem sind Publikationen im PDF-Format nicht empfehlenswert, da nicht garantiert werden kann, dass später keine Versionskonflikte mit aktuellen PDF-Viewern auftreten. Dafür schreitet die Entwicklung des Formates zu schnell voran.

Was benötigt wird, ist ein zukunftssicheres Format, ein Format, was auch in mehreren Jahren noch lesbar ist. Idealerweise sollte es portabel und ohne eine spezielle Applikation betrachtbar sein.

Hier bietet sich XML an. XML ist ein universelles Datenformat für die Veröffentlichung und den Austausch von strukturierten Dokumenten zwischen unterschiedlichen Systemen (vgl. Kapitel 2.2.3). Die große Stärke von XML ist, dass inhaltliche Strukturen in einer beliebigen Tiefe verschachtelt werden können, sodass auch komplexe Hierarchien jeder Art repräsentiert werden können. Das Konzept der Ilmedia-Schnittstelle verdeutlicht nachfolgende Abbildung.



**Abbildung 36: Schnittstelle zur Bibliothek der TU Ilmenau**

In einer Schnittstelle dieser Form spielt XML die Rolle eines Zwischenspeichers. Publikationen werden in XML exportiert und in eine Warteschlange abgelegt, bis sie geprüft werden. Diese Publikationen in XML werden nun der Bibliothek beziehungsweise dem Dienst Ilmedia zur Verfügung gestellt. Alle für die Publikation relevanten Informationen befinden sich in diesem Dokument. Ein Bibliotheksmitarbeiter kann daraufhin die Informationen des XML-Dokuments auslesen und die Veröffentlichung prüfen.

Diese Art eines Zwischenspeichers erhöht die Verfügbarkeit des Dienstes. Denn für den Nutzer, dessen Dokument in der Warteschlange angestellt wird, scheint der Dienst immer erreichbar zu sein, auch wenn dies nicht immer der Fall ist. Der Dienst entnimmt dann nacheinander die Dokumente aus der Warteschlange und bearbeitet sie. Denkbar ist auch eine parallele Verarbeitung der XML-Dokumente.

Damit eine im Alumni-Portal registrierte Publikation in den Katalog der Universitätsbibliothek eingetragen werden kann, sind bestimmte Informationen notwendig. Dazu zählen diverse Angaben zu den Autoren der Veröffentlichung und zu den Informationen der Publikation selbst. Damit die exportierten Daten nicht nur für die Bibliothek der TU Ilmenau interessant sein könnten, wurde ein universelles Daten-Schema entworfen. Dieses enthält zusätzliche Informationen, die für die Katalogisierung nicht notwendig sind, aber dazu führt, dass dieses Schema später noch für einen anderen Zweck eingesetzt werden kann.

Nachfolgend stellt eine Auflistung die Informationen des Schemas dar. Die von der Bibliothek der TU Ilmenau benötigten Informationen sind durch einen Unterstrich gekennzeichnet.

- Autoren
  - Verantwortlicher Autor
    - Name
    - Email
    - Telefonnummer
    - Adresse
  - Sonstige Autoren
- Sonstige beteiligte Personen
- Informationen zur Veröffentlichung
  - Titel
  - Medientyp
  - Datum der Einreichung
  - Datum der Veröffentlichung
  - Worin wurde es veröffentlicht
  - Verleger
  - URL zu der Veröffentlichung
  - Kurzfassung der Veröffentlichung
  - Anzahl der Seiten
  - ISBN Nummer
  - Auflage
  - Referenzen

Dieses Schema enthält alle möglichen Angaben verschiedenster Publikationen. Aus diesem Grund bleiben bei manchen Publikationen diverse Felder leer. Ein wissenschaftliches Papier hat weder einen Verleger, eine ISBN Nummer noch eine Auflage. Diese Angaben beziehen sich eher auf ein Buch. Durch die allgemeine Gestaltung deckt dieses Schema die meisten Publikationen ab.

### 5.2.4. Zusammenfassung

Das voran gegangene Kapitel hat erarbeitet, dass die beste Realisierung der Ilmedia-Schnittstelle durch das Konzept einer MOM erreicht wird. MOM integriert Systeme, indem Nachrichten zwischen den beteiligten Systemen ausgetauscht werden. Als Nachrichten-Format wurde XML ausgewählt. Die im Alumni-Portal registrierten Publikationen werden als XML-Dokument exportiert und der Bibliothek beziehungsweise dem Dienst Ilmedia übermittelt. Dieser prüft die Informationen der XML-Dokumente und fügt die Publikationen in den Katalog der Bibliothek ein.

## **5.3. Alumni-Datenbank**

### **5.3.1. Allgemeines zur Alumni-Datenbank**

Die TU Ilmenau führt eine Datenbank aller Studenten beziehungsweise ehemaliger Studenten in Ilmenau. Die sogenannte Alumni-Datenbank ist das Kernstück des Alumni-Netzwerkes. Sie ermöglicht den Kontakt zu den Alumni herzustellen und zu pflegen. Bisher sind etwa 6300 Absolventen in der Alumni-Datenbank erfasst.

### **5.3.2. Nutzen der Alumni-Datenbank**

Das Alumni-Portal kann durch den Dienst der Alumni-Datenbank eindeutig prüfen, ob ein Nutzer des Portals ehemaliger TU Ilmenau Student war oder nicht. Das Alumni-Portal nutzt die Alumni-Datenbank, um die Nutzer bei der Registrierung im Alumni-Portal als Studenten der Technischen Universität zu verifizieren.

### **5.3.3. Konzeption der AlumniDB-Schnittstelle**

Nachfolgend wird derselbe Weg zur Auswahl einer geeigneten Integrationstechnologie wie bei der Ilmedia-Schnittstelle verfolgt.

Bevor eine konkrete Integrationstechnologie ausgewählt werden kann, muss geklärt werden, wie die zu integrierenden Systeme gekoppelt werden und auf welcher Ebene die Integration ansetzt.

Ebenso wie bei der Ilmedia-Schnittstelle werden bei dieser Integration mit dem Alumni-Portal und der Absolventen-Datenbank der TU Ilmenau nur zwei Anwendungen miteinander verbunden. Die geeignetste Integrationstopologie ist die Punkt-zu-Punkt-Integration.

Da alle benötigten Informationen in einer zentralen Datenbank liegen, scheint die Datenintegration zunächst als beste Lösung. Eine Integration auf Datenebene mittels JDBC



oder ODBC ist aber nicht anwendbar. Demgegenüber stehen der Datenschutz und das Recht des Alumni auf informationelle Selbstbestimmung. Die Daten in der Alumni-Datenbank der TU Ilmenau unterliegen dem Datenschutz. Das bedeutet, dass Alumni-Portal darf ohne eine Zustimmung desjenigen auf keine Daten in der Datenbank zugreifen. Eine Datenintegration ist dadurch nicht möglich.

Eine Datenintegration ist aber auch nicht zwingend notwendig, denn es reicht, wenn ein Dienst auf Seiten der Alumni-Datenbank den Alumni bestätigt. Dazu müssen einzelne Daten aus der Registrierung übermittelt werden, die mit den in der Alumni-Datenbank gespeicherten Angaben verglichen werden. Tritt eine Übereinstimmung ein, so kann diese Person als Alumni der Technischen Universität Ilmenau bestätigt werden. Dadurch muss auf der Server-Seite eine Funktionalität realisiert sein, die dies prüft. Ist Anwendungslogik auf der Server-Seite vorhanden, die integriert werden soll, so ist die Funktionsintegration die einzige Art der Anwendungsintegration, die diese vorhandene Funktionalität wiederverwendbar beziehungsweise nutzbar machen kann. Deshalb ist für die Alumni-Datenbank die Funktionsintegration die sinnvollste Art der Anwendungsintegration.

Die Vielzahl an Integrationstechnologien für die Funktionsintegration macht es notwendig, die Anwendung weiteren Kriterien zu unterziehen (vgl. Kapitel 2.3.3). Die Ergebnisse unterstützen die Entscheidung einer konkreten Integrationstechnologie.

Da die Verifizierung während der Registrierung erfolgt, ist es wichtig, dass eine schnelle Antwort seitens der Alumni-Datenbank erfolgt. Der erfolgreiche Abschluss der Registrierung erfordert eine positive Rückmeldung der Alumni-Datenbank. Aus diesem Sachverhalt lässt sich ableiten, dass eine synchrone Kommunikation mit der zu integrierenden Anwendung notwendig ist. Der Dienst seitens des Alumni-Portals wartet bis die Alumni-Datenbank geantwortet hat. In dieser Zeit ist der Dienst blockiert und kann seine Arbeit nicht fortsetzen.

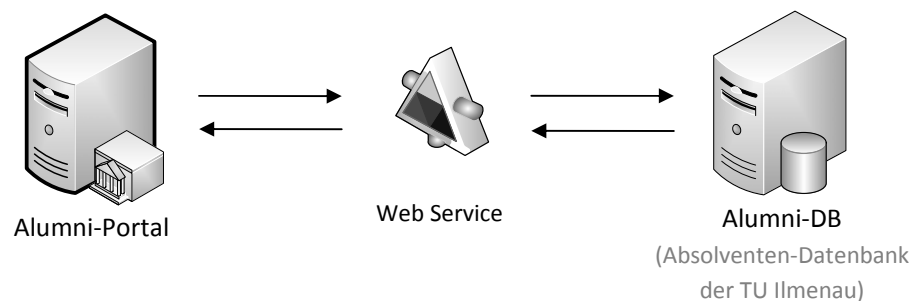
Nachdem das Kommunikationsmodell festgelegt ist, stellt sich die Frage nach dem Dienstaufrufmodell und dem Integrationsgegenstand.

Da bei dieser Art der Integration eine synchrone Kommunikation vorherrscht, ist ein nachrichten-basierter Dienstaufruf nicht zwingend erforderlich. Ein API-basierter Dienstaufruf ist ebenfalls anwendbar.

Die zu realisierende Schnittstelle soll den Zugriff auf die Alumni-Datenbank kapseln und trotzdem eine Antwort liefern, ob ein Nutzer des Portals Alumni oder wissenschaftlicher Mitarbeiter ist. Diese Funktionalität soll als Dienst verfügbar sein. Deshalb liegt ein dienstorientiertes Integrationsparadigma vor.

Betrachtet man nun die oben erarbeiteten Kriterien und die Abbildung 27 in Kapitel 2.3.4, so ist wieder eine Entscheidung zwischen einer MOM und einem Web Service als Integrationstechnologie zu treffen. Für die AlumniDB-Schnittstelle wurde ein Web Service als Integrationstechnologie ausgewählt. Ein Web Service setzt besser das dienstorientierte Integrationsparadigma um, als dies eine MOM tut.

Nachdem die Integrationstechnologie ausgewählt wurde, kann die AlumniDB-Schnittstelle umgesetzt werden. Nachfolgende Abbildung verdeutlicht die AlumniDB-Schnittstelle.



**Abbildung 37: Schnittstelle zur Alumni-DB**

Auf die Alumni-Datenbank der TU Ilmenau wird nicht direkt, sondern über einen Web Service zugegriffen. In Umgebungen in denen ein direkter Zugriff nicht möglich oder nicht gewollt ist, sind Web Services eine Möglichkeit, um trotzdem bestimmte Dienste in Anspruch zu nehmen. Die Server-Seite des Web Services kümmert sich um die Implementierung der Schnittstelle, sie ist der Client-Seite nicht bekannt. Die Client-Seite nutzt lediglich die Schnittstelle beziehungsweise den dadurch zur Verfügung gestellten Dienst. Der Web Service stellt wohldefinierte Methoden bereit, über die auf die Datenbank oder ähnliches zugegriffen werden kann. Dadurch werden die Daten gekapselt und Implementierungsdetails verborgen.

Bezogen auf das Web-Service-Rollenmodell aus Kapitel 2.2.6 fehlt in diesem Szenario die Rolle des Dienstvermittlers beziehungsweise des Dienstverzeichnisses. Wenn sich

Dienstnutzer und Dienstanbieter kennen und sich vorher über Syntax und Semantik ihrer Interaktion einigen, ist dieser auch nicht notwendig (vgl. Abbildung 14, Kapitel 2.2.6).

Damit ein Nutzer des Alumni-Portals eindeutig als Alumni identifiziert werden kann, müssen dem Web Service bestimmte Informationen übermittelt werden. Diese werden genutzt, um den Nutzer in der Absolventen-Datenbank zu finden. Dazu genügen der Name, der Vorname, das Geburtsdatum, der Geburtsort und die Matrikelnummer, die mit den Angaben in der Datenbank verglichen werden. Da das Alumni-Portal auch für wissenschaftliche Mitarbeiter der TU Ilmenau genutzt werden kann, die nicht notwendigerweise in Ilmenau studiert haben müssen, reichen die bisherigen Angaben nicht aus. Deshalb geben Mitarbeiter statt ihrer Matrikelnummer ihre Sozialversicherungsnummer an. Mit diesen sechs Informationen können sowohl die Alumni als auch die Mitarbeiter der Universität eindeutig identifiziert werden. Als Rückgabe des Web Services genügt ein Wahrheitswert, der mit *true* oder *false* den Nutzer des Alumni-Portals als Alumni oder wissenschaftlichen Mitarbeiter ausweist.

Nachfolgende Abbildung illustriert den AlumniDB-Web-Service.

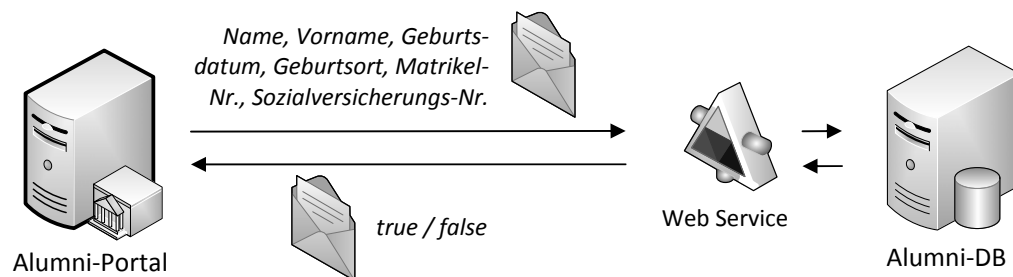


Abbildung 38: AlumniDB-Web-Service

### 5.3.4. Zusammenfassung

Das voran gegangene Kapitel hat erarbeitet, dass die beste Realisierung der AlumniDB-Schnittstelle durch einen Web Service erreicht wird. Ein Web Service ist eine geeignete Möglichkeit, die Vertrauenswürdigkeit der Alumni-Daten zu wahren und trotzdem die gewünschte Funktionalität zu realisieren. Dazu werden dem Web Service der Name, der Vorname, das Geburtsdatum, der Geburtsort, die Matrikelnummer und die Sozialversicherungsnummer des Nutzers übergeben. Dieser prüft die Daten und

entscheidet mit true oder false, ob der Nutzer ein Alumni oder wissenschaftlicher Mitarbeiter der TU Ilmenau ist.

## 6. Implementierung der Schnittstellen

In diesem Kapitel werden prototypische Implementierungen der Schnittstellen zu der Bibliothek und der Absolventen-Datenbank der TU Ilmenau entwickelt.

### 6.1. Ilmedia-Schnittstelle

Die Schnittstelle zur Bibliothek der TU Ilmenau wird, wie im vorherigen Kapitel erarbeitet, durch eine XML-Schnittstelle realisiert. Dazu wird zuerst eine konkrete XML-Schema-Definition (XSD) erarbeitet, auf dessen Grundlage konkrete Implementierungen entwickelt werden, die ein XML-Dokument gemäß dem XSD erzeugen. Die Entwicklung eines XSLT-Stylesheets für die benutzerfreundliche Darstellung des XML-Dokumentes schließt das Kapitel ab.

#### 6.1.1. Entwicklung der Ilmedia-XSD

##### 1.) Entwicklungsumgebung

Aufgrund des Textformats lassen sich XML-Dokumente, DTDs und XSDs oder XSL- und XSLT-Dateien im Prinzip mit jedem Texteditor erstellen und bearbeiten. Doch es gibt eine ganze Reihe von komfortablen XML- und Stylesheet-Editoren, die dem Anwender viel Arbeit abnehmen. Für die folgende Entwicklung wurde *XMLSpy* ausgewählt.

##### *XMLSpy*

Die Firma ALTOVA bietet mit *XMLSpy*<sup>1</sup> eine mächtige Entwicklungsumgebung, die die Arbeit mit XML enorm erleichtert. *XMLSpy* unterstützt kontextsensitive Elementlisten und Attribut-Inspektoren und sorgt durch den integrierten XML-Prozessor dafür, dass die Wohlgeformtheit und auch die Gültigkeit (vgl. Kapitel 2.2.3) schon während der Entwicklung jederzeit geprüft werden kann. Insbesondere der Entwurf von DTDs oder XML Schemas wird sehr erleichtert durch graphische Ansichten, in denen die Baumstruktur eines XML-Dokumentes aufgebaut und geprüft werden kann. Zudem ist *XMLSpy* in der Lage, DTDs oder XSDs aus bestehenden XML-Dokumenten zu generieren

---

<sup>1</sup> Link: [http://www.altova.com/de/produkte/xmlspy/xml\\_editor.html](http://www.altova.com/de/produkte/xmlspy/xml_editor.html)

oder DTDs in XSDs zu transformieren. Zwar ist *XMLSpy* nicht frei verfügbar, doch ALTOVA bietet eine vollwertige Trial-Version an, die 30 Tage kostenlos genutzt werden kann.

### 2.) *Ilmedia-XSD*

XSD ist eine komplexe Schemasprache zum Definieren von XML-Dokumentstrukturen. Anders als bei den klassischen DTD wird die Struktur in Form eines XML-Dokuments beschrieben (vgl. Kapitel 2.2.3).

Das Kapitel 5.3.3 hat bereits ein erstes Schema der zu exportierenden Daten erarbeitet. Dieses gilt es zu verfeinern und als XSD zu notieren. Dazu müssen die Elemente strukturiert und mit Datentypen versehen werden. XSD unterscheidet zwischen einfachen und komplexen Datentypen. Einfache Typen dürfen weder Kindelemente enthalten noch Attribute besitzen. Nachfolgende Auflistung zeigt die wichtigsten einfachen XSD-Datentypen.

- `xsd:string`
- `xsd:decimal`
- `xsd:integer`
- `xsd:float`
- `xsd:boolean`
- `xsd:date`
- `xsd:time`
- `xsd:anyURI`

Im Gegensatz zu den einfachen Typen bieten komplexe Datentypen die Möglichkeit, Kindelemente sowie Attribute zu definieren.

Die nachfolgende Abbildung illustriert die der XSD zu Grunde liegende Hierarchie. In runden Klammern ist der Datentyp des jeweiligen Elements angegeben. In den eckigen Klammern sind die Kardinalitäten der Kindelemente notiert.

- Veröffentlichungen (Komplexer Typ) [1..n]
  - Veröffentlichung (Komplexer Typ) [1..1]
    - Autoren (Komplexer Typ) [1..1]
      - Verantwortlicher Autor (Komplexer Typ) [1..1]
        - Name (Einfacher Typ - xsd:string)
        - Email (Einfacher Typ - xsd:string)
        - Telefonnummer (Einfacher Typ - xsd:string)
        - Adresse (Einfacher Typ - xsd:string)
      - Sonstige Autoren (Komplexer Typ) [1..5]
        - Autor (Einfacher Typ - xsd:string)
    - Sonstige beteiligte Personen (Komplexer Typ) [1..5]
      - Person (Einfacher Typ - xsd:string)
    - Informationen zur Veröffentlichung (Komplexer Typ) [1..1]
      - Titel (Einfacher Typ - xsd:string)
      - Medientyp (Einfacher Typ - xsd:string)
      - Datum der Einreichung (Einfacher Typ - xsd:date)
      - Datum der Veröffentlichung (Einfacher Typ - xsd:date)
      - Worin wurde es veröffentlicht (Einfacher Typ - xsd:string)
      - Verleger (Einfacher Typ - xsd:string)
      - URL zu der Veröffentlichung (Einfacher Typ - xsd:anyURI)
      - Kurzfassung der Veröffentlichung (Einfacher Typ - xsd:string)
      - Anzahl der Seiten (Einfacher Typ - xsd:string)
      - ISBN Nummer (Einfacher Typ - xsd:string)
      - Auflage (Einfacher Typ - xsd:string)
      - Referenzen (Komplexer Typ) [1..30]
        - Referenz (Einfacher Typ - xsd:string)
    - Kommentar zur Veröffentlichung (Einfacher Typ - xsd:string)

Abbildung 39: Hierarchie der Ilmedia-XSD

Das Wurzelement der Ilmedia-XSD ist mit `publications` bezeichnet. `publications` ist ein komplexer Datentyp, der aus einer Sequenz von Kindelementen besteht. Die Kindelemente sind eine oder mehrere `publication`-Elemente.

```
<xsd:element name="publications">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="publication" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

**Listing 3: publications-Element**

Dadurch ist die Anforderung erfüllt, dass das exportierte Format mehrere Publikationen enthalten kann. Das Element `publication` ist ebenfalls ein komplexer Typ, bestehend aus einer Sequenz der Kindelemente `authors`, `persons`, `publication_informations` und `comment`. Die Elemente innerhalb einer `sequence` müssen in der angegebenen Reihenfolge auftreten. Jedes dieser Elemente kann nicht oder mehrfach auftreten und ist gekennzeichnet durch die Attribute `minOccurs` und `maxOccurs`. Falls kein `occurs`-Attribut vorhanden ist, wird der Default-Wert 1 verwendet.

```
<xsd:element name="publication">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="authors" />
      <xsd:element ref="persons" />
      <xsd:element ref="publication_informations" />
      <xsd:element ref="comment" minOccurs="0" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

**Listing 4: publication-Element**

Die Kindelemente `authors`, `persons` und `publication_informations` sind wieder komplexe Typen, wo hingegen das optionale Element `comment` ein einfacher `xsd:string` Typ ist.

```
<xsd:element name="comment" type="xsd:string" />
```

**Listing 5: comment-Element**



Das Element `authors` ist wieder unterteilt in die Kindelemente `corresponding_author` und `author`.

```
<xsd:element name="authors">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="corresponding_author"/>
      <xsd:element ref="author" minOccurs="0"
                    maxOccurs="5"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="corresponding_author">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="name"/>
      <xsd:element ref="email" minOccurs="0"/>
      <xsd:element ref="phone" minOccurs="0"/>
      <xsd:element ref="address" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="author" type="xsd:string"/>
<xsd:element name="name" type="xsd:string"/>
<xsd:element name="email" type="xsd:string"/>
<xsd:element name="phone" type="xsd:string"/>
<xsd:element name="address" type="xsd:string"/>
```

Listing 6: authors-Element

Der `corresponding_author` ist ein komplexer Typ, der aus einer Sequenz von `xsd:string` Typen besteht. Neben dem `corresponding_author` besteht der Typ `authors` zusätzlich aus maximal fünf `author`-Elementen. `author`-Elemente sind vom Typ `xsd:string`.

Das Element `persons` enthält eine Sequenz von maximal fünf `xsd:string` Typen.

```
<xsd:element name="persons">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="person" minOccurs="0"
                    maxOccurs="5"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="person" type="xsd:string"/>
```

Listing 7: persons-Element

Das letzte komplexe Element ist das Element `publication_informations`. Dieses besteht aus einer Sequenz der Kindelemente `title`, `mediatype`, `submission_date`, `published_date`, `published_in`, `publisher`, `url`, `abstract`, `pages`, `isbn`, `edition` und `references`.

```
<xsd:element name="publication_informations">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="title"/>
      <xsd:element ref="mediatype"/>
      <xsd:element ref="submission_date" minOccurs="0"/>
      <xsd:element ref="published_date"/>
      <xsd:element ref="published_in" minOccurs="0"/>
      <xsd:element ref="publisher" minOccurs="0"/>
      <xsd:element ref="url"/>
      <xsd:element ref="abstract"/>
      <xsd:element ref="pages" minOccurs="0"/>
      <xsd:element ref="isbn" minOccurs="0"/>
      <xsd:element ref="edition" minOccurs="0"/>
      <xsd:element ref="references" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

**Listing 8: publication\_informations-Element**

Optionale Elemente sind durch `minOccurs="0"` gekennzeichnet. Alle Kindelemente bis auf `references` sind `xsd:string`-, `xsd:anyURI`- oder `xsd:date`-Typen. Lediglich `references` ist wieder ein komplexer Typ bestehend aus maximal 30 `reference`-Elementen vom Typ `xsd:string`.

```
<xsd:element name="url" type="xsd:anyURI"/>
<xsd:element name="title" type="xsd:string"/>
<xsd:element name="mediatype" type="xsd:string"/>
<xsd:element name="publish_date" type="xsd:date"/>
<xsd:element name="published_in" type="xsd:string"/>
<xsd:element name="publisher" type="xsd:string"/>
<xsd:element name="abstract" type="xsd:string"/>
<xsd:element name="pages" type="xsd:string"/>
<xsd:element name="isbn" type="xsd:string"/>
<xsd:element name="edition" type="xsd:string"/>
<xsd:element name="submission_date" type="xsd:date"/>

<xsd:element name="references">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="reference" minOccurs="0"
                    maxOccurs="30"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="reference" type="xsd:string"/>
```

Listing 9: Einfache Elemente

Im Überblick sieht die IImedia-XSD folgendermaßen aus. Die Darstellung ist mit *XMLSpy* erstellt worden.

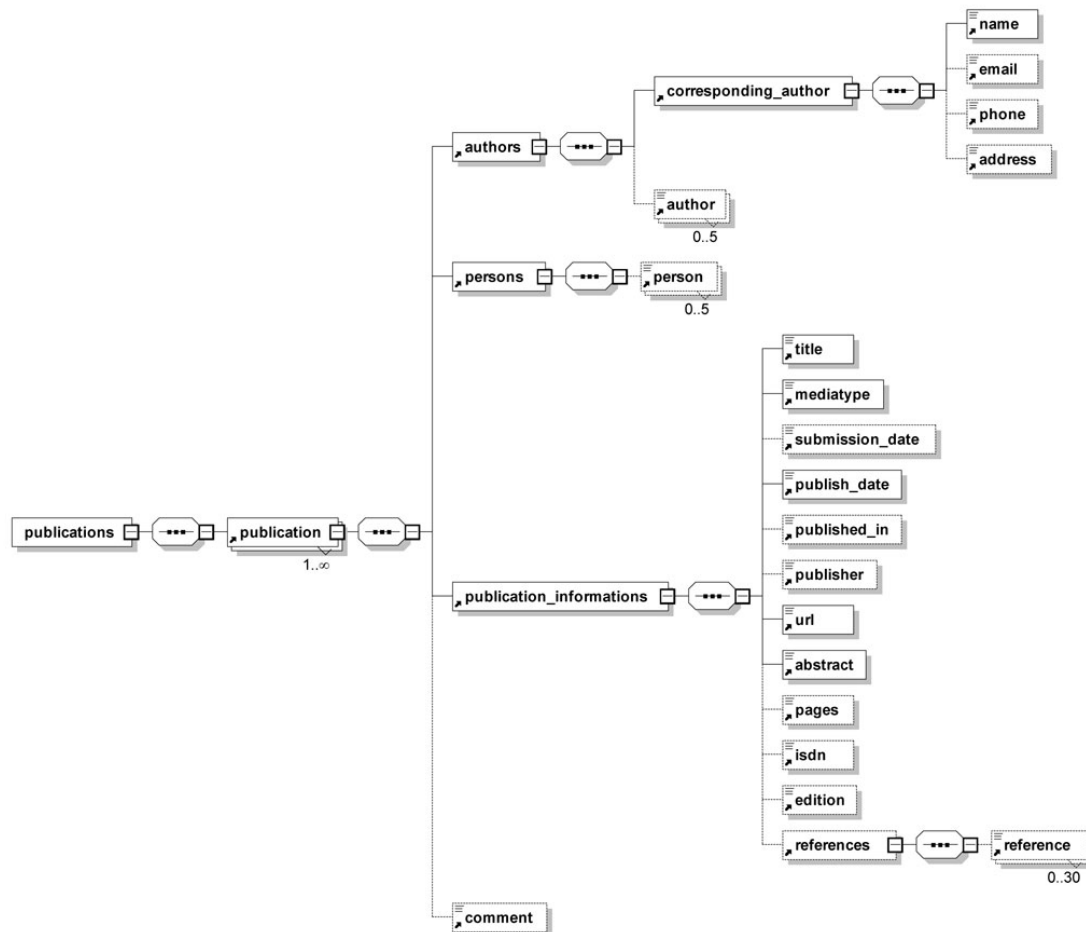


Abbildung 40: Graphische Darstellung der Ilmedia-XSD in XMLSpy

Die Zahlen unter den Elementen geben die Kardinalität an. Eine gepunktete Umrandung eines Elements visualisiert die Optionalität des Elements.

Dieses XML-Schema kann nun umgesetzt werden, indem das zu exportierende XML-Dokument diesem Schema folgt. Die Implementierung erfolgt in PHP, weil das Alumni-Portal ebenfalls in PHP realisiert wird und so der Quellcode direkt eingebunden werden kann.

## 6.1.2. Implementierung in PHP

### 1.) PHP

PHP wurde 1995 von Rasmus Lerdorf entwickelt und war ursprünglich eine Sammlung von Perl-Skripten. Da das WWW Ende der 90er Jahre stark wuchs, bestand großer Bedarf an Scriptsprachen, die in der Lage seien sollten dynamische Web-Seiten zu realisieren. PHP wurde für die Web-Entwicklung mit der Zeit populärer als der vorherige De-facto-Standard Perl. Im Juli 2004 erschien PHP in der Version 5. Erstmals wurde durch viele hinzugefügte Sprachkonstrukte ein ernsthaftes objektorientiertes Programmieren möglich. [10]

### 2.) XAMPP

*XAMPP* ist eine Zusammenstellung von freier Software und ermöglicht das einfache Installieren und Konfigurieren des Web-Servers *Apache* mit der Datenbank *MySQL* beziehungsweise *SQLite* und den Scriptsprachen *Perl* und *PHP*. *XAMPP* ist über die *ApacheFriends*-Website<sup>1</sup> in einer Linux-, einer Mac-, einer Solaris- und auch in einer Windows-Variante verfügbar. Nicht nur die umfassende Ausstattung dieses Pakets macht es so interessant, sondern weitere Eigenschaften sprechen für *XAMPP*. So ist *XAMPP* kostenlos und als der Versuch des *ApachesFriends*-Teams zu verstehen, die Entwicklung freier Software zu unterstützen. [36]

### 3.) Entwicklungsumgebung

Um in PHP zu entwickeln, ist nicht unbedingt eine Entwicklungsumgebung erforderlich. Prinzipiell ist für einfache PHP-Applikationen ein beliebiger Text-Editor völlig ausreichend. Doch PHP-Entwicklungsumgebungen bieten zur Verwaltung umfangreicherer PHP-Projekte Funktionalitäten, die einfache Editoren nicht leisten können. Dadurch erhöht sich die Produktivität und der Komfort beim Entwickeln und Verwalten von PHP-Applikationen.

Für die Entwicklung der Imedia-Schnittstelle wurde die Entwicklungsumgebung *Zend Studio 5.5*<sup>2</sup> ausgewählt. Neben einer vollständigen dynamischen Codeergänzung, Dokumentationsgenerierung, wahlweisen Anbindung an die Versionsverwaltungssysteme

---

<sup>1</sup> Link: <http://www.apachefriends.org>

<sup>2</sup> Link: [http://www.zend.com/de/products/zend\\_studio](http://www.zend.com/de/products/zend_studio)

*Subversion* und *CVS*<sup>1</sup> sowie *FTP*<sup>2</sup>- und *MySQL*-Integration bietet das *Zend Studio* auch einen integrierten Debugger, der PHP selbst fehlt. Zwar ist das *Zend Studio 5.5* nicht frei verfügbar, doch Zend bietet eine vollwertige Trial-Version an, die 30 Tage kostenlos genutzt werden kann.

#### **4.) XML Verarbeitung mit PHP**

Der wachsenden Bedeutung von XML trägt PHP 5 mit einer von Grund auf überarbeiteten Unterstützung für die XML-Verarbeitung Rechnung. PHP 4 verfügte in der Standardinstallation nur über eine Unterstützung für die ereignisbasierte XML-Verarbeitung. PHP 5 hingegen bietet in seiner Standardinstallation mehrere Erweiterungen für die Verarbeitung von XML-Dokumenten. [37]

Für die Implementierung der Ilmedia-Schnittstelle wurde die DOM-Erweiterung der SAX-Erweiterung vorgezogen, da SAX kein Schreibzugriff erlaubt und DOM Lese- und Schreibzugriffe auf beliebige XML-Elemente ermöglicht.

Die DOM-Erweiterung implementiert den DOM-Standard und ermöglicht eine baumbasierte und objektorientierte Verarbeitung von XML-Dokumenten. DOM fasst jeden Bestandteil eines XML-Dokumentes als einen Knoten auf. Das Dokument selbst, seine Elemente sowie deren Attribute und Inhalte werden durch Objekte entsprechender Knotenklassen repräsentiert.

Die DOM-Implementierung von PHP 5 umfasst 29 Klassen mit insgesamt 360 Methoden. Die wichtigsten sind in der nachfolgenden Tabelle aufgelistet.

<b>Klasse</b>	<b>Elternklasse oder Schnittstelle</b>	<b>Aufgabe</b>
DOMNode		Basisklasse der verschiedenen Knotenklassen eines DOM-Baumes.
DOMDocument	DOMNode	Bildet den Einstiegspunkt des DOM-Baumes.
DOMElement	DOMNode	Repräsentiert ein XML-Element und

---

<sup>1</sup> Concurrent-Versions-System, Software-System zur Versionsverwaltung, Link: <http://www.nongnu.org/cvs/>

<sup>2</sup> File-Transfer-Protocol, RFC: <http://tools.ietf.org/html/rfc959>

		bietet Methoden für den Zugriff auf XML-Elemente
DOMAttr	DOMNode	Repräsentiert ein Attribut eines XML-Elementes
DOMCharacterData	DOMNode	Repräsentiert den Character-Data-Teil eines XML-Elementes
DOMText	DOMCharacterData	Repräsentiert textuelle Inhalte eines XML-Dokuments
DOMNodeList	Traversable	Repräsentiert eine Menge von Knoten
DOMXPath		Ermöglicht das Formulieren von XPath-Anfragen
DOMException	Exception	Ausnahmeklasse

Tabelle 6: Die wichtigsten Klassen des DOM in PHP 5 (Quelle: [37])

### 5.) Implementierung der XML Schnittstelle

Das Erzeugen eines XML-Dokumentes durch die direkte Verwendung der DOM-Klassen, ist recht aufwändig. Aus diesem Grund sind die DOM-Klassen durch Vererbung erweitert worden.

Die Klassen `Publication`, `Authors`, `Persons`, `PublicationInformations`, `CorrespondingAuthor` und `References` erweitern die Klasse `DOMElement` um die für die Elemente spezifischen Methoden `create____()` und `set____(____)`. Damit wird das Aufrufen von `DOMDocument::createElement()` und `DOMNode::appendChild()` abgenommen.

Nachfolgendes Listing zeigt die Klasse `Publication`, die von der Klasse `DOMElement` abgeleitet ist und zusätzlich die Methoden `createAuthors()`, `createPersons()`, `createPublicationInformations()` und `setComment()` implementiert.

```
1 <?php
2
3 class Publication extends DOMElement
4 {
5     private $document;
6
7     public function __construct(DOMDocument $document)
8     {
9         parent::__construct('publication');
10        $this->document=$document;
11    }
12
13
14    public function createAuthors()
15    {
16        $authors = new Authors($this->document);
17        $this->appendChild($authors);
18
19        return $authors;
20    }
21
22    public function createPersons()
23    {
24        $persons = new Persons($this->document);
25        $this->appendChild($persons);
26
27        return $persons;
28    }
29
30    public function createPublicationInformations()
31    {
32        $publication_informations = new PublicationInformations
33                                   ($this->document);
34        $this->appendChild($publication_informations);
35
36        return $publication_informations;
37    }
38
39    public function setComment($comment)
40    {
41        $this->appendChild
42        (
43            $this->document->createElement
44            (
45                'comment', $comment
46            )
47        );
48    }
49 }
50 ?>
```

**Listing 10: Die Klasse Publication**



Dem Konstruktor der Klasse wird ein Objekt vom Typ `DOMDocument` übergeben, welches als private Variable `document` gespeichert wird. Dieses Objekt muss den nachfolgenden Elementen übergeben werden, damit diese die Methode `createElement()` aufrufen können. `DOMDocument::createElement()` erzeugt ein neues Objekt der Klasse `DOMElement`. Die drei `create____()` Methoden erzeugen jeweils neue `DOMElemente`, welche dem `publication-Element` durch die Methode `appendChild()` angefügt werden. Die Methode `setComment($comment)` erzeugt hingegen ein Objekt namens `comment` vom Typ `DOMText` und übergibt diesem den Wert `$comment`.

Die Klassen `Authors`, `Persons`, `PublicationInformations`, `CorrespondingAuthor` und `References` folgen demselben Aufbau und befinden sich auf der beigefügten CD.

Die Klasse `Publications` erweitert die Klasse `DOMDocument` und bietet mit ihrer Methode `createPublication()` die Möglichkeit, neue `publication-Elemente` zu erzeugen.

```
1 <?php
2
3 class Publications extends DOMDocument
4 {
5     private $publications;
6
7     public function __construct()
8     {
9         parent::__construct('1.0','ISO-8859-1');
10
11         $this->appendChild($this->createProcessingInstruction
12 ('xml-stylesheet', 'href="publication.xslt" type="text/xsl"));
13
14         $this->publications = $this->appendChild
15         (
16             $this->createElement('publications')
17         );
18
19         $this->formatOutput = TRUE;
20     }
21
22     public function createPublication()
23     {
24         $publication = new Publication($this);
25         $this->publications->appendChild($publication);
26
27         return $publication;
28     }
29 }
30 ?>
```

Listing 11: Die Klasse `Publications`

Dem Konstruktor der Klasse werden die gewünschte Version des XML-Standards und die zu verwendende Zeichenkodierung übergeben. Zusätzlich werden noch Verarbeitungshinweise in Form eines XSLT-Stylesheets dem Dokument angehängt. Die Zeile `$this->formatOutput = TRUE` bewirkt, dass das XML-Dokument wohlformatiert wird. Das bedeutet, dass jedes Element in eine neue Zeile geschrieben und gemäß der Hierarchie der Elemente eingerückt wird.

Durch die Verwendung der Klassen `Publications`, `Publication`, `Authors`, `Persons`, `PublicationInformations`, `CorrespondingAuthor` und `References` kann nun ein XML-Dokument erstellt werden.

Das Listing 12 benutzt die erweiterten Klassen beziehungsweise deren Methoden um die folgende, beispielhafte Publikation (Abbildung 41) als XML-Dokument, gemäß der in Kapitel 6.1.1 erarbeiteten XSD, zu exportieren.

- Veröffentlichung
  - Autoren
    - Verantwortlicher Autor
      - Name: *Mustermann*
      - Email: *Max*
      - Telefonnummer: *0170/1234567*
      - Adresse: *Musterstraße 1, 98765 Musterstadt*
    - Sonstige Autoren
      - Autor: *Hans Meier*
      - Autor: *Maria Müller*
  - Sonstige beteiligte Personen
    - Person: *Erika Mustermann*
  - Informationen zur Veröffentlichung
    - Titel: *Beispielveröffentlichung*
    - Medientyp: *Paper*
    - Datum der Einreichung: *2007-03-30*
    - Datum der Veröffentlichung: *2007-06-30*
    - Worin wurde es veröffentlicht: *Musterbuch*
    - Verleger: *Musterverlag*
    - URL zu der Veröffentlichung:  
*<http://www.musterurl.de/musterveröffentlichung.pdf>*
    - Kurzfassung der Veröffentlichung: *Dieses Paper analysiert...*
    - Anzahl der Seiten: *8*
    - ISBN Nummer: *978-1234567890*
    - Auflage: *1*
    - Referenzen
      - Referenz: Maria Müller, Erika Mustermann:  
*'Musterreferenz 1', 2003.*
      - Referenz: Maria Müller, Hans Meier: *'Musterreferenz 2', 2006.*
      - Referenz: Erika Mustermann, Hans Meier:  
*'Musterreferenz 3', 2007.*
  - Kommentar zur Veröffentlichung: *Dies ist eine Beispiel-Veröffentlichung*

Abbildung 41: Daten der zu exportierenden Publikation

```

1  <?php
2
3  require_once('Publication2XML.php');
4
5  $publications = new Publications();
6
7  $publication = $publications->createPublication();
8
9  $authors = $publication->createAuthors();
10 $persons = $publication->createPersons();
11 $publication_informations = $publication->
    createPublicationInformations();
12 $comment = $publication->setComment
    ("Dies ist eine Beispiel-Veröffentlichung");
13
14 $corresponding_author = $authors->createCorrespondingAuthor();
15 $authors->setAuthor("Hans Meier");
16 $authors->setAuthor("Maria M&uuml;ller");
17
18 $corresponding_author->setName("Max Mustermann");
19 $corresponding_author->setEmail("Max.Mustermann@email.de");
20 $corresponding_author->setPhone("0170/1234567");
21 $corresponding_author->setAddress
    ("Musterstraße 1, 98765 Musterstadt");
22
23 $persons->setPerson("Erika Mustermann");
24
25 $publication_informations->setTitle("Beispielveröffentlichung");
26 $publication_informations->setMediatype("Paper");
27 $publication_informations->setSubmissionDate("2007-03-30");
28 $publication_informations->setPublishDate("2007-06-30");
29 $publication_informations->setPublishedIn("Musterbuch");
30 $publication_informations->setPublisher("Musterverlag");
31 $publication_informations->setUrl
    ("http://www.musterurl.de/musterveröffentlichung.pdf");
32 $publication_informations->setAbstract
    ("Dieses Paper analysiert...");
33 $publication_informations->setPages("8");
34 $publication_informations->setIsbn("978-1234567890");
35 $publication_informations->setEdition("1");
36 $references = $publication_informations->createReferences();
37
38 $references->setReference
    ("Maria Müller, Erika Mustermann: 'Musterreferenz 1', 2003.");
39 $references->setReference
    ("Maria Müller, Hans Meier: 'Musterreferenz 2', 2006.");
40 $references->setReference
    ("Erika Mustermann, Hans Meier: 'Musterreferenz 3', 2007.");
41
42 $publications->save('publication.xml');
43
44 echo nl2br(htmlentities($publications->saveXML()));
45
46 ?>

```

Listing 12: Beispielcode zur Erzeugung einer Publikation in XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<?xml-stylesheet href="publication.xslt" type="text/xsl"?>

<publications>
  <publication>

    <authors>
      <corresponding_author>
        <name>Max Mustermann</name>
        <email>Max.Mustermann@email.de</email>
        <phone>0170/1234567</phone>
        <address>Musterstraße 1, Musterstadt</address>
      </corresponding_author>
      <author>Hans Meier</author>
      <author>Maria Müller</author>
    </authors>

    <persons>
      <person>Erika Mustermann</person>
    </persons>

    <publication_informations>
      <title>Beispielveröffentlichung</title>
      <mediatype>Paper</mediatype>
      <submission_date>2007-03-30</submission_date>
      <publish_date>2007-06-30</publish_date>
      <published_in>Musterbuch</published_in>
      <publisher>Musterverlag</publisher>
      <url>http://www.musterurl.de/musterveröffentlichung.pdf</url>
      <abstract>Dieses Paper erarbeitet...</abstract>
      <pages>8</pages>
      <isbn>978-1234567890</isbn>
      <edition>1</edition>
      <references>
        <reference>
          Maria Müller, Erika Mustermann: 'Musterreferenz1',
          2003.
        </reference>
        <reference>
          Maria Müller, Hans Meier: 'Musterreferenz2', 2006.
        </reference>
        <reference>
          Erika Mustermann, Hans Meier: 'Musterreferenz3', 2007.
        </reference>
      </references>
    </publication_informations>

    <comment>Dies ist eine Beispiel-Veröffentlichung</comment>

  </publication>
</publications>
```

Listing 13: publication.xml

### 6. 1. 3. Entwicklung des Imedia-XSLT-Stylesheets

XSL-Transformations (XSLT) sind Teil der Extensible-Stylesheet-Language (XSL). Sogenannte XSLT-Stylesheets erlauben die Definition von Umwandlungsregeln. Spezielle XSLT-Prozessoren lesen XSLT-Stylesheets ein und transformieren die XML-Dokumente nach den Stylesheet-Regeln in das gewünschte Ausgabeformat (vgl. Kapitel 2.2.3).

In der obigen Implementierung ist im Konstruktor der Klasse `Publications` folgende Zeile enthalten.

```
11 $this->appendChild($this->createProcessingInstruction  
    ('xml-stylesheet', 'href="publication.xslt" type="text/xsl"));
```

Listing 14: Verarbeitungsanweisung in PHP

Diese fügt im erzeugten XML-Dokument folgende Verarbeitungsanweisung hinzu.

```
2 <?xml-stylesheet href="publication.xslt" type="text/xsl"?>
```

Listing 15: Verarbeitungsanweisung in XML

Das erzeugte XML-Dokument ist durch diese Anweisung mit einem XSLT-Stylesheet verknüpft. Wird das XML-Dokument geöffnet, so wird im selben Ordner nach dem XSLT-Stylesheet gesucht, um das XML-Dokument gemäß den Vorgaben zu formatieren.

Wurde nun die im Alumni-Portal registrierte Publikation als XML-Dokument exportiert, so bietet XSLT die Möglichkeit, das Dokument formatiert zu präsentieren. Ein Mitarbeiter der Bibliothek muss dafür keine spezielle Applikation besitzen, ein Web-Browser genügt.

Listing 16 zeigt das der Formatierung zu Grunde liegende XSLT-Stylesheet. Es handelt sich lediglich um einen Auszug, das komplette Stylesheet befindet sich auf der beigefügten CD.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:output method="html"/>

<xsl:template match="/">
  <html>
    <head>
      <title>Publikations-Liste</title>
      <link type="text/css"
        href="style.css"
        rel="stylesheet"/>
    </head>
    <body>
      <xsl:apply-templates />
    </body>
  </html>
</xsl:template>

<xsl:template match="publication">
  
  <span id="headline">Publikation Nr.
    <xsl:number/></span>
  <xsl:apply-templates />
</xsl:template>

<xsl:template match="authors">
  <div id="border">
    <h3><u>Autoren:</u></h3>
    <xsl:apply-templates />
  </div>
</xsl:template>

<xsl:template match="corresponding_author">
  <i><xsl:value-of select="name"/> (Verantwortlicher Autor)</i>
  <xsl:apply-templates select="email"/>
  <xsl:apply-templates select="phone"/>
  <xsl:apply-templates select="address"/>
</xsl:template>

...

```

Listing 16: Ilmedia-XSLT-Stylesheet

Eine Transformation besteht aus einer Reihe von einzelnen Transformationsregeln, die Templates genannt werden. Ein Template besitzt ein auf XPath<sup>1</sup> basierendes Pattern, das beschreibt, für welche Knoten es gilt und einen Inhalt, der bestimmt, wie das Template seinen Teil des Zielbaums erzeugt.

<sup>1</sup> Die XML Path Language (XPath) ist eine vom W3C-Konsortium entwickelte Anfragesprache um Teile eines XML-Dokumentes zu adressieren, W3C Recommendation: <http://www.w3.org/TR/xpath20/>

Auf diese Weise schließt das folgende Beispiel alle `authors`-Elemente in `<div id="border">`-Tags ein und fügt die unterstrichene Überschrift „*Autoren:*“ hinzu. `<xsl:apply-templates/>` bedeutet, dass der XSLT-Prozessor nach weiteren Transformationsregeln für die Kindelemente von `authors` suchen soll.

```
<xsl:template match="authors">
  <div id="border">
    <h3><u>Autoren:</u></h3>
    <xsl:apply-templates />
  </div>
</xsl:template>
```

**Listing 17: XSL-Transformation des `authors`-Elements**

Wird nun das XML-Dokument mit einem Web-Browser geöffnet, so liest der integrierte XSLT-Prozessor das XSLT-Stylesheet ein und transformiert das XML-Dokument in das gewünschte Ausgabeformat. Die Abbildung 42 zeigt die Ausgabe des XML-Dokuments im Web-Browser gemäß dem XSLT-Stylesheet.



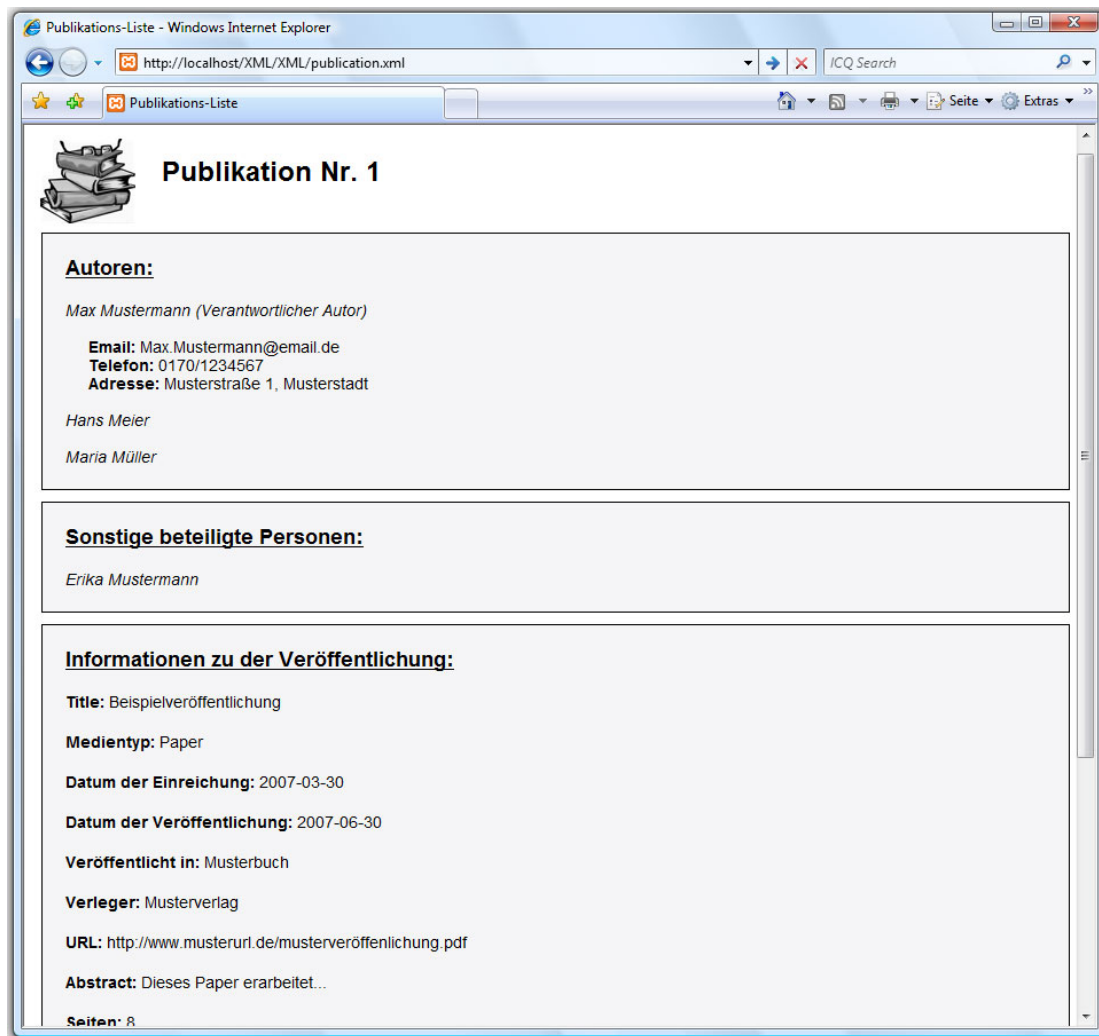


Abbildung 42: Darstellung der publication.xml gemäß dem XSLT-Stylesheet

## 6.2. AlumniDB-Schnittstelle

Die Schnittstelle zur Absolventen-Datenbank der TU Ilmenau wird, wie im vorherigen Kapitel erarbeitet, durch einen Web Service realisiert. Zuerst wird eine Web-Service-Beschreibung (WSDL) erstellt, aus der anschließend prototypische Implementierungen abgeleitet werden können. Der Web Service wird in den Programmiersprachen Java und PHP sowohl server- wie auch client-seitig umgesetzt.

### 6.2.1. Entwicklung der AlumniDB-WSDL

#### 1.) AlumniDB-WSDL

Der AlumniDB-Web-Service wird in der Beschreibungssprache WSDL formuliert. WSDL ist eine Metasprache, mit deren Hilfe die angebotenen Funktionen, Daten, Datentypen und Austauschprotokolle eines Web Services beschrieben werden. Es werden im Wesentlichen die Operationen definiert, die von außen zugänglich sind, sowie die Parameter und Rückgabewerte dieser Operationen. Die Spezifikation eines Web Services in WSDL ist ein XML-Dokument und besteht aus den Elementen `definitions`, `types`, `message`, `portType`, `binding` und `service` (vgl. Kapitel 2.2.6).

Die Alumni-Datenbank wird genutzt, um den Nutzer bei der Registrierung im Portal als Alumni der TU Ilmenau zu verifizieren. Um einen Nutzer als Alumni oder wissenschaftlichen Mitarbeiter zu identifizieren, werden dem AlumniDB-Web-Service der Name, der Vorname, das Geburtsdatum, der Geburtsort und die Matrikelnummer beziehungsweise die Sozialversicherungsnummer übergeben, die mit den Angaben in der Alumni-Datenbank verglichen werden. Die Rückgabe des Web Services ist ein Wahrheitswert, der mit *true* oder *false* den Nutzer des Alumni-Portals als Alumni oder wissenschaftlichen Mitarbeiter ausweist.

Nachfolgend werden alle Elemente der AlumniDB-WSDL erläutert.

Das Element `definitions` stellt das Wurzelement eines jeden WSDL-Dokumentes dar. Es werden vorwiegend Namensräume definiert.

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions
  targetNamespace="http://alumniDB.de/wsdl/AlumniDB"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://alumniDB.de/wsdl/AlumniDB" name="AlumniDB">
```

Listing 18: definitions-Element

Der `targetNamespace` spezifiziert einen eindeutigen Bezeichner für den mit diesem Dokument erzeugten Namensraum. Alle hier neu definierten Elemente gehören automatisch zu diesem Namensraum. `tns` stellt ein zusätzliches Präfix für diesen Namensraum dar. Darüber hinaus werden die Namensraum-Präfixe `xsd` und `soap` eingeführt. Sie referenzieren die XML-Schema-Spezifikation sowie die WSDL-Erweiterung für SOAP.

In der AlumniDB-WSDL sind zwei logische Nachrichtentypen definiert. Das sind zum einen der `SearchAlumniRequest` und zum anderen der `SearchAlumniResponse`.

```
<message name="SearchAlumniRequest">
  <part name="Name" type="xsd:string"/>
  <part name="Firstname" type="xsd:string"/>
  <part name="Birthdate" type="xsd:string"/>
  <part name="Birthplace" type="xsd:string"/>
  <part name="MatriculationNumber" type="xsd:string"/>
  <part name="NationalInsuranceNumber" type="xsd:string"/>
</message>

<message name="SearchAlumniResponse">
  <part name="Result" type="xsd:boolean"/>
</message>
```

Listing 19: message-Element

Die bei dem Web Service ankommende Nachricht `SearchAlumniRequest` enthält die Parameter `Name`, `Firstname`, `Birthdate`, `Birthplace`, `MatriculationNumber` und `NationalInsuranceNumber`. Alle sechs Parameter sind vom Typ `xsd:string`. Die vom Web Service ausgehende Nachricht `SearchAlumniResponse` besitzt einen Parameter namens `Result` vom Typ `xsd:boolean`.

Das `portType`-Element dient der Beschreibung von Ports. Es gibt an, welche Funktion oder Operation über ein Port erbracht wird. Um welchen Port es sich handelt, wird mit `name` angegeben. In Abhängigkeit vom Port-Typ sind hier die Elemente `input` und `output` vorhanden, in denen entsprechend die empfangene und die gesendete Nachricht als `message` angegeben ist.

```
<portType name="AlumniDBPortType">
  <operation name="SearchAlumni">
    <input name="input" message="tns:SearchAlumniRequest"/>
    <output name="output" message="tns:SearchAlumniResponse"/>
  </operation>
</portType>
```

**Listing 20: portType-Element**

Der Web Service erbringt die Funktion `SearchAlumni`. Der Aufruf der Funktion erfolgt über den Eingangsport `input` mit der Nachricht `SearchAlumniRequest`. Das Ergebnis des Aufrufs wird über den Ausgangsport `output` in `SearchAlumniResponse` übermittelt. Die Reihenfolge des `input`- und `output`-Elements bestimmt den Typ der Operation. Es handelt sich in diesem Fall um eine Request-Response-Operation (vgl. Tabelle 2, Kapitel 2.2.6).

Das `binding`-Element definiert die Bindung an ein konkretes Protokoll. Es legt fest, wie genau eine im Element `portType` definierte Operation über das Netzwerk transportiert wird. Für die präzise Beschreibung konkreter Nachrichtenformate wurde WSDL erweitert. Für die SOAP-Erweiterung wurde im Element `definitions` das Namensraum-Präfix `soap` definiert. Die wichtigsten Elemente sind `soap:binding`, `soap:operation` und `soap:body`.

```
<binding name="AlumniDBBinding" type="tns:AlumniDBPortType">
  <soap:binding style="rpc"
    transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="SearchAlumni">
    <soap:operation/>
    <input name="input">
      <soap:body use="literal"
        namespace="http://alumniDB.de/wsdl/AlumniDB"/>
    </input>
    <output name="output">
      <soap:body use="literal"
        namespace="http://alumniDB.de/wsdl/AlumniDB"/>
    </output>
  </operation>
</binding>
```

Listing 21: binding-Element

Das `binding`-Element referenziert den Port-Typ mit dem Attribut `type`. `AlumniDBPortType` soll auf das SOAP-Protokoll abgebildet werden. `soap:binding` legt das Kommunikationsmodell `rpc` fest und definiert HTTP als Transportprotokoll. Bei `rpc` als Kommunikationsmodell werden Parameter und Rückgabewerte in eine vorgegebene XML-Struktur eingebettet. Die Angaben innerhalb von `operation` erweitern die abstrakten Angaben des gleichnamigen Elements in `portType`. Für jede dort definierte Operation muss hier eine gleichnamige Operation spezifiziert werden. Für jede Nachricht der Operation ist festzulegen, wo und in welcher Form sie in der SOAP-Message erscheinen soll. Das Attribut `use` legt fest, wie die Daten kodiert werden sollen.

Das Element `service` beschreibt einen oder mehrere Ports. Der Port `AlumniDBPort` verknüpft die zuvor definierte Bindung `AlumniDBBinding` mit der Netzwerkadresse und definiert so den Endpunkt.

```
<service name="AlumniDBService">
  <port name="AlumniDBPort" binding="tns:AlumniDBBinding">
    <soap:address
      location="http://localhost/AlumniDB/AlumniDB"/>
  </port>
</service>
```

Listing 22: service-Element

## 2.) Ableitung von prototypischen Implementierungen

Nachdem der zu realisierende Web Service in WSDL spezifiziert wurde, können nun konkrete Implementierungen abgeleitet werden.

Das Alumni-Portal wird in der Programmiersprache PHP entwickelt. Aus diesem Grund erfolgt eine Implementierung des AlumniDB-Web-Service in PHP. Zudem erfolgt eine weitere Implementierung in Java, weil Java mit seiner Web-Plattform sehr ausgereifte Werkzeuge für die Entwicklung eines Web Services bietet. Außerdem sind Java-Application-Server *State-of-the-Art* als Ablaufumgebung von server-seitiger Anwendungslogik.

In PHP und Java werden sowohl Implementierungen für die Client- als auch für die Server-Seite entwickelt.

## 6. 2. 2. Implementierung des Web Services in Java

### 1.) JavaEE 5

Die Programmiersprache Java wurde 1995 von *Sun Microsystems* veröffentlicht und verbreitete sich rasch. Zunächst war Java nur für Desktop-Anwendungen konzipiert. Als sich das Internet mehr und mehr verbreitete und Bedarf an interaktiven Web Applikationen vorhanden war, wurde eine Aufteilung der Java Plattform vorgenommen. Die alte Plattform der Desktop-Anwendungen bekam den Namen *Standard-Edition*, während man die neue Plattform, welche auf Web-Applikationen ausgerichtet wurde, *Enterprise-Edition* nannte. Im Mai 2006 erschien schließlich die *Java Enterprise Edition 5 (JavaEE5)*.

*JAX-WS 2.0*<sup>1</sup> ist ein neues Framework der *JavaEE5*, mit der Web Services erstellt werden können. Damit soll die alte Schnittstelle *JAX-RPC 1.1* abgelöst werden. Während mit älteren Java-Spezifikationen das Aufsetzen eines Web Services eine komplexere Angelegenheit war, wird die Verteilungsproblematik von *JAX-WS* weitestgehend gekapselt und die Anwendungsentwicklung stark vereinfacht.

---

<sup>1</sup> Java API for XML - Web Services, JAX-WS 2.0 hieß ursprünglich JAX-RPC 2.0. Da der Name *Remote Procedure Call* allerdings nicht mehr zutreffend und irreführend ist, wurde der Name zu JAX-WS 2.0 geändert, JAX-WS 1.0 gibt es somit nicht.

## 2.) Entwicklungsumgebung

Um einen Web Service zu implementieren, muss eine geeignete Entwicklungsumgebung ausgesucht werden. Dafür wurde *NetBeans* in der Version 5.5 ausgewählt. *NetBeans* eignet sich hervorragend für die Entwicklung von Web Services, da es im Bündel mit dem *Sun Java System Application Server 9.0* und diversen anderen Tools zur Verfügung steht. Alle notwendigen Komponenten sind schon integriert, was viel Konfigurationsaufwand erspart.

Eine ebenfalls kostenlose Alternative zu *NetBeans* ist die mächtige Entwicklungsumgebung *Eclipse*. Für viele Projekte ist *Eclipse* vielleicht die bessere Wahl, doch in diesem Fall hat *NetBeans* die Nase vorn.

Um an eine vorkonfigurierte Entwicklungsumgebung zu kommen, muss das oben angesprochene *JavaEE5 Tools Bundle* auf der *NetBeans-Homepage*<sup>1</sup> heruntergeladen werden. Neben *NetBeans* und dem Application-Server sind auch andere hilfreiche Tools dabei, wie beispielsweise *JavaDOC* und das *NetBeans Enterprise Pack*. Einzige Voraussetzung ist ein vorinstalliertes *Java SDK*<sup>2</sup> 1.5, welches ebenfalls dort verfügbar ist.

## 3.) AlumniDB-Web-Service bereitstellen (ohne WSDL)

Um einen Web Service zu erstellen, muss auf der Server-Seite eine Klasse erstellt werden, die den Web Service implementiert.

Dazu wird in *NetBeans* eine Web-Applikation erstellt. Wichtig dabei ist, dass ein *Java EE5 Projekt* erstellt wird, da der Service auf der *Enterprise Edition 5* beruht. Im Folgenden wird der *Sun Java System Application Server 9* verwendet, prinzipiell kann aber jeder Application-Server verwendet werden, der die *JavaEE5*-Spezifikation implementiert. Soll das Projekt in einen Web-Container laufen, genügt auch der *Apache Tomcat 5.5*. Dieser ist ebenfalls im *JavaEE5 Tools Bundle* enthalten.

---

<sup>1</sup> Link: <http://www.netbeans.info/downloads/>

<sup>2</sup> Java-SE-Development-Kit

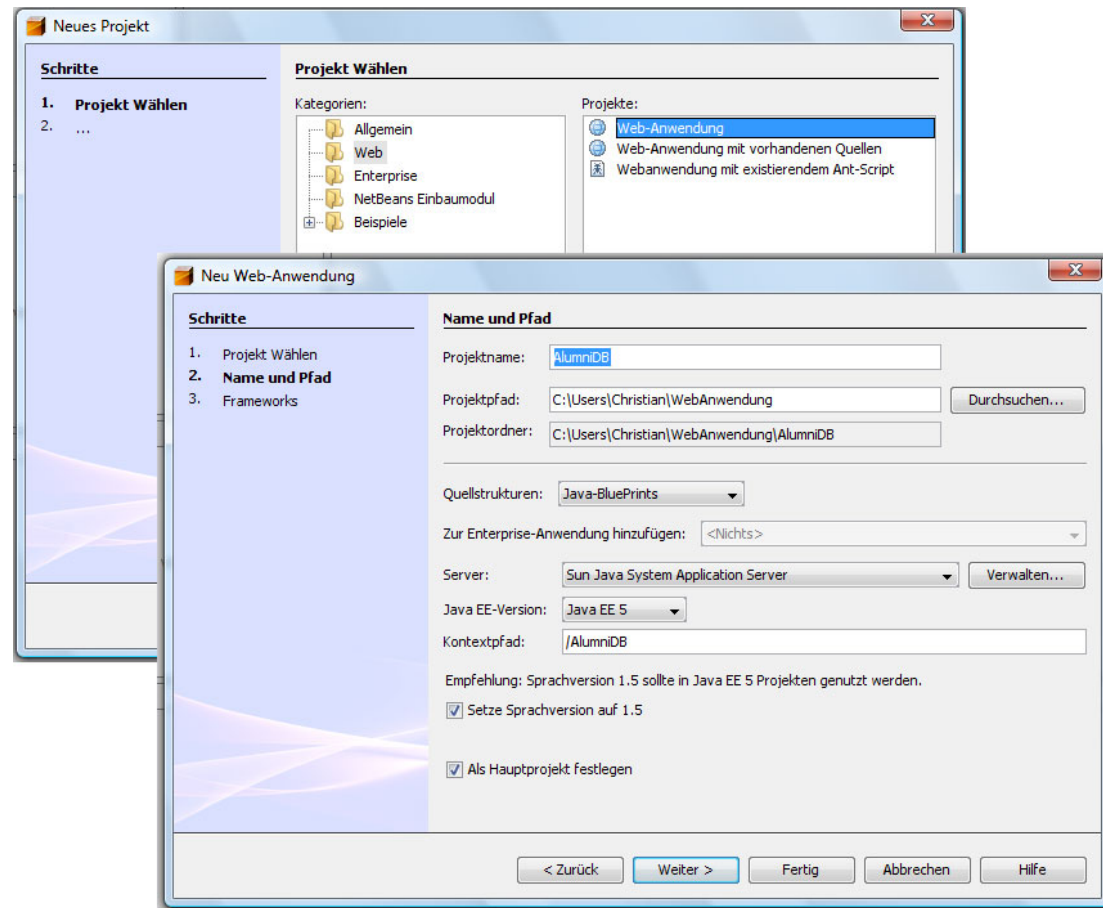


Abbildung 43: Erstellen einer Web-Anwendung in NetBeans

Nun wird eine neue Java-Klasse erstellt, welche die Web-Service-Implementierung repräsentiert. Die Schnittstelle des Web Services wird mit Annotationen deklariert. Diese liegen im Paket `javax.jws`.

- `@WebService()` spezifiziert die Klasse als Web Service
- `@WebMethod()` kennzeichnet die Methoden der Service-Schnittstelle
- `@WebParam()` kennzeichnet die Parameter, die der Service als Eingabeparameter benötigt.

Nach der Codierung muss noch die Auslieferung des Projekts in den Web-Container vorgenommen werden. Die *JavaEE5*-Laufzeitumgebung generiert nun das WSDL-Dokument des Servers und das SOAP-Skeleton, welches die Service-Aufrufe entgegennimmt. Standardmäßig reagiert der Application-Server auf Anfragen an den Port 8080.



```
1 package org.me.alumniDBServer;
2
3 import javax.jws.WebMethod;
4 import javax.jws.WebParam;
5 import javax.jws.WebService;
6
7
8 @WebService()
9 public class AlumniDB
10 {
11     /*
12     * Die Web Service Implementierung
13     * Durch Annotationen wird die Service-Schnittstellendeklariert
14     */
15
16     @WebMethod()
17     public boolean SearchAlumni (
18         @WebParam(name = "Name")
19         String Name,
20         @WebParam(name = "Firstname")
21         String Firstname,
22         @WebParam(name = "Birthdate")
23         String Birthdate,
24         @WebParam(name = "Birthplace")
25         String Birthplace,
26         @WebParam(name = "MatriculationNumber")
27         String MatriculationNumber,
28         @WebParam(name = "NationalInsuranceNumber")
29         String NationalInsuranceNumber)
30     {
31         /*
32         * Die fachliche Logik
33         * Suchen des Nutzers in der Alumni-Datenbank
34         */
35     }
36 }
```

Listing 23: AlumniDB-Web-Service in Java (ohne WSDL)

Nachdem sichergestellt ist, dass der Server läuft, kann der AlumniDB-Web-Service getestet werden. Dazu bietet der *Sun Java System Application Server 9* ein webbasiertes Testmodul an. Dieses kann direkt aus *NetBeans* heraus oder über das *Administration Web Interface*<sup>1</sup> aufgerufen werden.

---

<sup>1</sup> Link: <http://localhost:4848/>

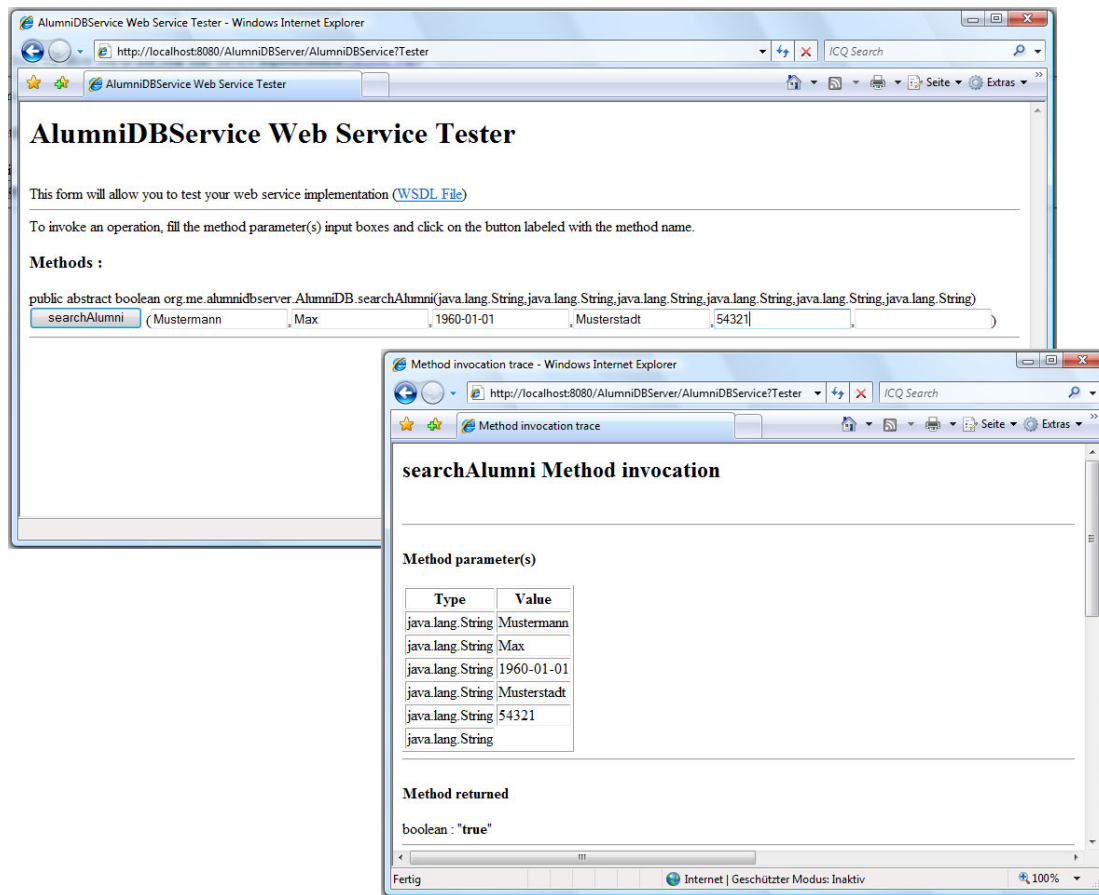


Abbildung 44: Web-Service-Tester des SJSAS 9

Aus den Informationen des WSDL-Dokuments kann ein Service-Nutzer nun erfahren, wie er mit dem Web Service interagieren kann. Das Testmodul ermöglicht einen einfachen Test, ohne dass eine Implementierung eines Web-Service-Clients notwendig ist. Der Web-Service-Tester ermöglicht es auch, die übertragenen SOAP-Nachrichten einzusehen (Listing 45 und 46).

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <soapenv:Envelope
   xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
   xmlns:ns1="http://alumniDBServer.me.org/">
3   <soapenv:Body>
4     <ns1:SearchAlumni>
5       <Name>Mustermann</Name>
6       <Firstname>Max</Firstname>
7       <Birthdate>1960-01-01</Birthdate>
8       <Birthplace>Musterstadt</Birthplace>
9       <MatriculationNumber>54321</MatriculationNumber>
10      <NationalInsuranceNumber> </NationalInsuranceNumber>
11    </ns1:SearchAlumni>
12  </soapenv:Body>
13 </soapenv:Envelope>
```

Abbildung 45: SOAP-Request

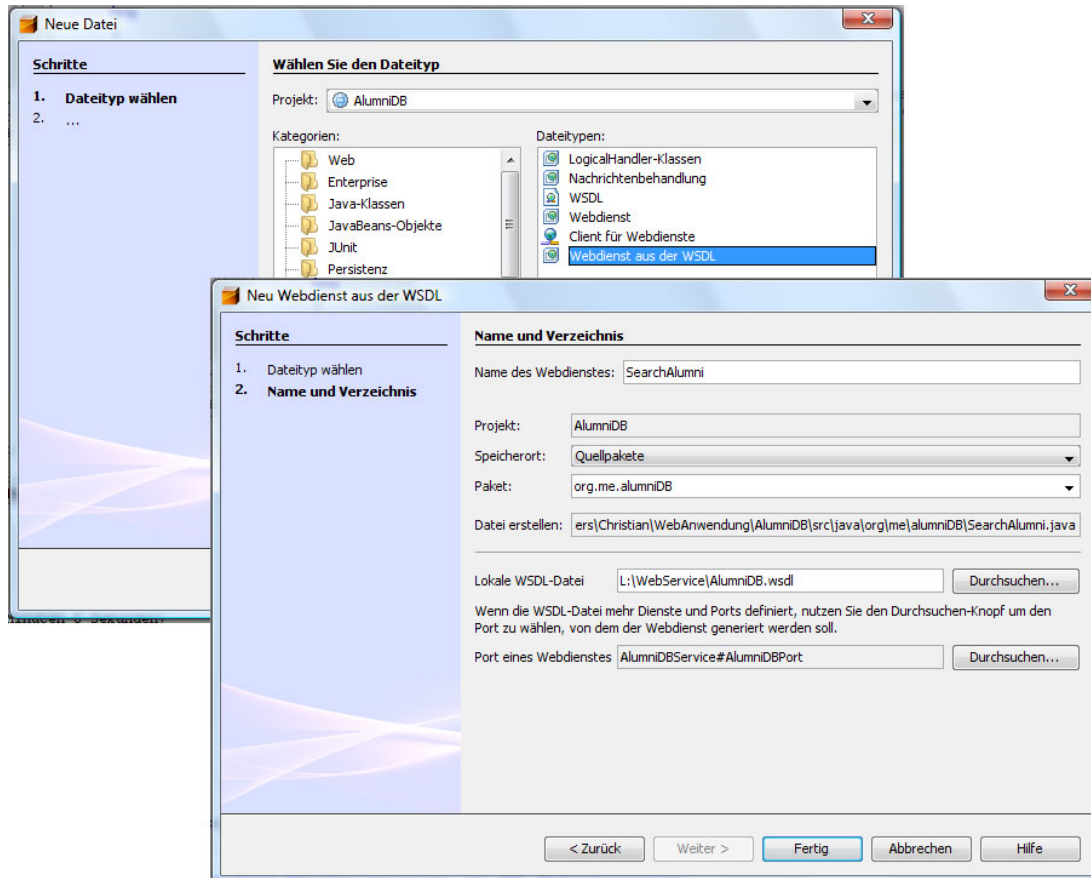
```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <soapenv:Envelope
   xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
   xmlns:ns1="http://alumniDBServer.me.org/">
3   <soapenv:Body>
4     <ns1:SearchAlumniResponse>
5       <return>true</return>
6     </ns1:SearchAlumniResponse>
7   </soapenv:Body>
8 </soapenv:Envelope>
```

Abbildung 46: SOAP-Response

Der obige AlumniDB-Web-Service wurde aus einer Java-Klasse erstellt. Aber auch der umgekehrte Fall ist möglich, indem aus einem WSDL-Dokument ein Web Service erzeugt wird.

#### 4.) AlumniDB-Web-Service bereitstellen (mit WSDL)

Um einen Web Service aus einem WSDL-Dokument zu erzeugen, wird in *NetBeans* eine neue Web-Applikation erstellt und anschließend durch den Menüpunkt *Web Service aus der WSDL* ein Web Service abgeleitet. Dazu muss der Ort des WSDL-Dokuments angegeben werden.



Danach muss noch die Datei `SearchAlumni.java` angepasst werden. *NetBeans* generiert aus der WSDL lediglich den Rumpf des Web Services, ohne jegliche Funktion. Die Anwendungslogik muss nun noch implementiert werden.

```
1 package org.me.alumniDB;
2
3 import javax.jws.WebService;
4
5 @WebService(
6     serviceName = "AlumniDBService",
7     portName = "AlumniDBPort",
8     endpointInterface =
9         "de.alumnidb.wsdl.alumnidb.AlumniDBPortType",
10    targetNamespace = "http://alumniDB.de/wsdl/AlumniDB",
11    wsdlLocation = "WEB-INF/wsdl/SearchAlumni/AlumniDB.wsdl")
12 public class SearchAlumni implements
13     de.alumnidb.wsdl.alumnidb.AlumniDBPortType
14 {
15     public SearchAlumni ()
16     {
17     }
18
19     public boolean searchAlumni (
20         String name,
21         String firstname,
22         String birthdate,
23         String birthplace,
24         String matriculationNumber,
25         String nationalInsuranceNumber)
26     {
27         /*
28          *   Die fachliche Logik
29          *   Suchen des Nutzers in der Alumni-Datenbank
30          */
31     }
```

Listing 24: AlumniDB Web Service in Java (mit WSDL)

### 5.) AlumniDB-Web-Service konsumieren (mit WSDL)

Als Konsument des Web Services wird eine Java-Desktop-Applikation implementiert. Dazu wird in *NetBeans* eine *Java Anwendung* erstellt, der Menüpunkt *Client für Webdienste* ausgewählt und das WSDL-Dokument übergeben. Für die Referenz muss ein Paketname angegeben werden, unter dem die Klassen dann verfügbar sind, als wären sie lokal auf dem System vorhanden.

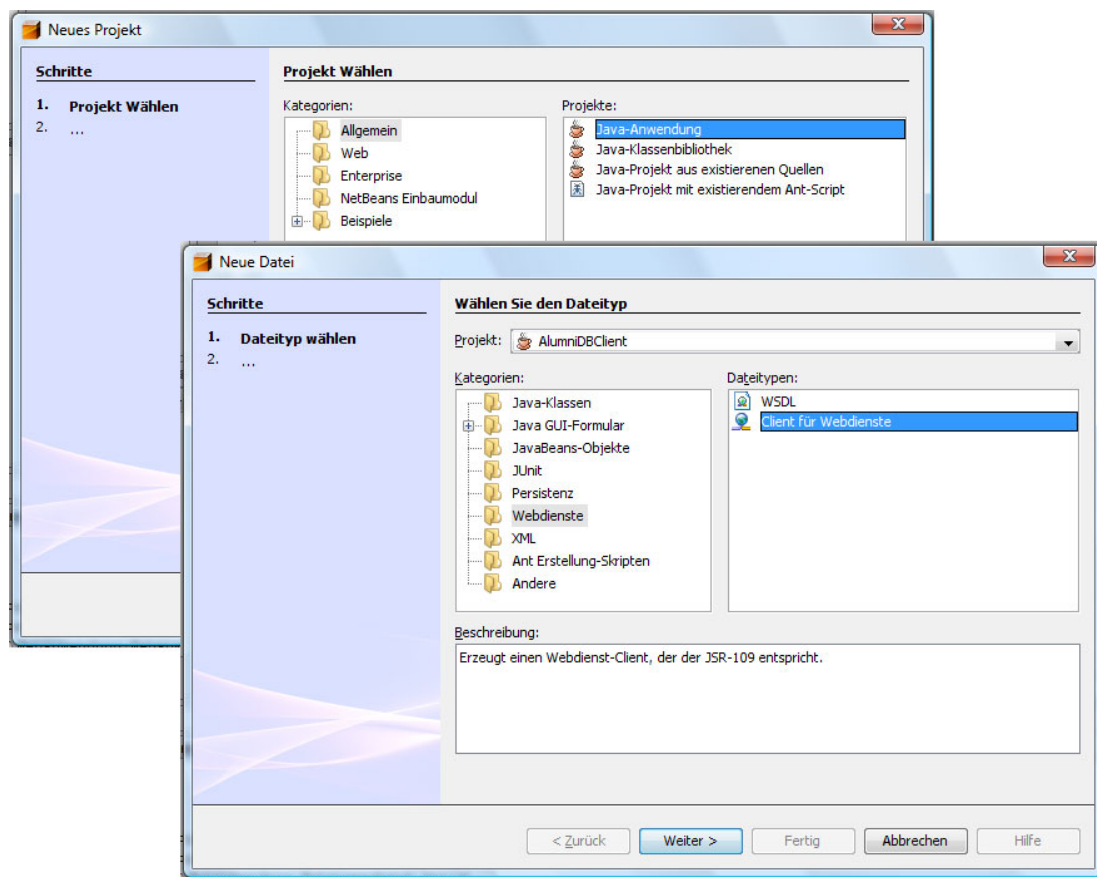


Abbildung 47: Web-Service-Client in Java

Anschließend muss noch die Datei `Main.java` angepasst werden. Sie soll den Web Service konsumieren.

```
1 public class Main
2 {
3     public static void main (String[] args)
4     {
5         try
6         {
7             org.me.alumniDBClient.AlumniDBService service = new
                        org.me.alumniDBClient.AlumniDBService ();
8             org.me.alumniDBClient.AlumniDB port =
                        service.getAlumniDBPort();
9
10            java.lang.String name = "Mustermann";
11            java.lang.String firstname = "Max";
12            java.lang.String birthdate = "1960-01-01";
13            java.lang.String birthplace = "Musterstadt";
14            java.lang.String matriculationNumber = "54321";
15            java.lang.String nationalInsuranceNumber = "";
16
17            boolean result = port.searchAlumni
                        (name, firstname, birthdate, birthplace,
                        matriculationNumber, NationalInsuranceNumber);
18            System.out.println ("Alumni in Datenbank: "+result);
19        }
20        catch (Exception ex)
21        {
22            System.out.println("Error: "+ex);
23        }
24    }
25 }
```

Listing 25: AlumniDB-Web-Service-Client in Java

## 6.2.3. Implementierung des Web Services in PHP

### 1.) PHP

PHP 5 bietet eine eingebaute Extension, um SOAP-Server oder SOAP-Clients zu erstellen. Es gibt zwar noch weitere SOAP-Implementierungen<sup>1</sup> für PHP 5, jedoch sind diese meist in PHP verfasst und somit von der Performanz einer in der Programmiersprache C geschriebenen Extension unterlegen.

In der standardmäßigen Installation von *XAMPP* ist die für die Entwicklung von Web Services wichtige SOAP-Extension deaktiviert. Diese muss nachträglich in der `php.ini` aktiviert werden.

```
658 ;extension=php_smtp.dll
659 ;extension=php_snmp.dll
660 extension=php_soap.dll
661 extension=php_sockets.dll
662 extension=php_sqlite.dll
663 ;extension=php_ssh2.dll
664 ;extension=php_stats.dll
```

Listing 26: `php.ini`

### 2.) Entwicklungsumgebung

Für die Entwicklung des Web Service in PHP wurde ebenfalls die Entwicklungsumgebung *Zend Studio* ausgewählt (vgl. Kapitel 6.1.2).

### 3.) AlumniDB-Web-Service bereitstellen (ohne WSDL)

Jede Funktion oder öffentliche Methode einer Klasse kann in PHP 5 über die Klasse `SOAPServer` als Web Service verfügbar gemacht werden.

Um einen SOAP-Server mit PHP 5 zu erstellen, gibt es zwei Möglichkeiten. Man kann einen SOAP-Server sowohl mit als auch ohne WSDL-Dokument erzeugen. Der Konstruktor der Klasse `SOAPServer` erwartet entweder den Pfad zu einer WSDL-Beschreibung des Web Services als ersten Parameter oder ein assoziatives Array, das Informationen wie die URL des Web Services enthält, als zweiten Parameter.

---

<sup>1</sup> z.B.: SOAPx4, eZ SOAP, Krysalis, NuSOAP, PEAR::SOAP



Listing 27 zeigt, dass der Wert `NULL` als erster Parameter sowie ein Array mit der Konfiguration des Web Services als zweiter Parameter übergeben wird.

```
1 <?php
2 class AlumniDB
3 {
4     public function SearchAlumni(
5         $Name,
6         $Firstname,
7         $Birthdate,
8         $Birthplace,
9         $MatriculationNumber,
10        $NationalInsuranceNumber)
11     {
12         /*
13          *   Die fachliche Logik
14          *   Suchen des Nutzers in der Alumni-Datenbank
15          */
16     }
17 }
18
19 try
20 {
21     $server = new SOAPServer
22     (
23         NULL,
24         array
25         (
26             'uri' => 'http://localhost/AlumniDB/SearchAlumni'
27         )
28     );
29
30     $server->setClass('AlumniDB');
31     $server->handle();
32 }
33
34 catch (SOAPFault $f)
35 {
36     print $f->faultstring;
37 }
38 ?>
```

Listing 27: AlumniDB Web Service in PHP (ohne WSDL)

Mit der Methode `setClass()` des `SOAPServer`-Objektes wird die Klasse festgelegt, deren öffentliche Methode über den Web Service verfügbar gemacht werden soll. Mit der Methode `handle()` wird der Web Service gestartet. Das bedeutet, dass eine HTTP-Anfrage an das Script des `SOAPServer`-Objektes als Anfrage an den Web Service aufgefasst und entsprechend verarbeitet wird. Die Fehler, die während der SOAP-Verarbeitung auftreten,

werden durch eine `SOAPFault`-Ausnahme repräsentiert und können entsprechend abgefangen werden.

#### 4.) AlumniDB-Web-Service bereitstellen (mit WSDL)

Bei der Erzeugung des `SOAPServer`-Objektes wird nun anstelle von `NULL` und dem assoziativen Parameter-Array der Pfad zur WSDL-Beschreibung angegeben. Für die HTTP-Anfrage an `http://localhost/AlumniDB/SOAPServer.php?wsdl` gibt das Script die WSDL-Beschreibung aus.

```
1 <?php
2 class AlumniDB {
3     public function SearchAlumni(
4         $Name,
5         $Firstname,
6         $Birthdate,
7         $Birthplace,
8         $MatriculationNumber,
9         $NationalInsuranceNumber)
10    {
11        /*
12         *   Die fachliche Logik
13         *   Suchen des Nutzers in der Alumni-Datenbank
14         */
15    }
16 }
17
18 try
19 {
20     $server = new SOAPServer('AlumniDB.wsdl');
21     $server->setClass('AlumniDB');
22     $server->handle();
23 }
24
25 catch (SOAPFault $f)
26 {
27     print $f->faultstring;
28 }
29 ?>
```

Listing 28: AlumniDB-Web-Service in PHP (mit WSDL)

#### 5.) AlumniDB-Web-Service konsumieren (mit WSDL)

Für die Nutzung eines Web Services bietet PHP die Klasse `SOAPClient` an. Ein Objekt dieser Klasse dient hierbei als Stellvertreter für den Web Service. Methodenaufrufe werden automatisch in eine SOAP-Message übersetzt und an den entsprechenden Web Service

weitergeleitet. Die SOAP-Response wird verarbeitet und als Ergebnis des Methodenaufrufes zurückgegeben.

Liegt für den zu verwendenden Web Service ein WSDL-Dokument vor, so ist dem Konstruktor der Klasse `SOAPClient` lediglich der Pfad oder die URL zu übergeben, wo das Dokument zu finden ist.

Listing 29 zeigt den Zugriff auf den AlumniDB-Web-Service über dessen WSDL-Beschreibung.

```
1 <?php
2 try
3 {
4     $client = new SOAPClient
5         ( 'http://localhost/AlumniDB/SOAPServerWSDL.php?wsdl' );
6     print $client->SearchAlumni
7         (
8             'Mustermann',
9             'Max',
10            '1969-10-18',
11            'Musterstadt',
12            '54321',
13            ''
14        );
15 }
16
17 catch (SOAPFault $f)
18 {
19     print $f->faultstring;
20 }
21 ?>
```

Listing 29: Web-Service-Client in PHP (mit WSDL)

Die Erzeugung eines `SOAPClient`-Objektes für die Verwendung eines Web Services, für den keine WSDL-Beschreibung vorliegt, ist um einiges aufwändiger. Die sonst aus dem WSDL-Dokument gewonnenen Informationen müssen dann dem Konstruktor in einem assoziativen Array übergeben werden.

```
1 <?php
2 try
3 {
4     $client = new SOAPClient(
5         NULL,
6         array
7         (
8             'location' => 'http://localhost/AlumniDB/SOAPServer.php',
9             'uri'       => 'http://localhost/AlumniDB/SearchAlumni',
10            'style'      => SOAP_RPC,
11            'use'        => SOAP_ENCODED
12        ));
13
14    print $client->SearchAlumni
15        ('Mustermann', 'Max', '1960-01-01', 'Musterstadt', '54321', '');
16
17 catch (SOAPFault $f)
18 {
19     print $f->faultstring;
20 }
21 ?>
```

**Listing 30: Web-Service-Client in PHP (ohne WSDL)**

Über die Methode `__getFunctions()` der Klasse `SOAPClient` können Informationen über die Funktionen des Web Services abgefragt werden. Hierzu gehören der Name der Funktion, Name und Typ der Parameter sowie der Typ des Rückgabewertes.

```
1 <?php
2     $client = new SOAPClient
3         ('http://localhost/AlumniDB/AlumniDB.wsdl');
4     print_r($client->__getFunctions());
5 ?>
```

**Listing 31: Abfragen von Web-Service-Informationen**

```
X-Powered-By: PHP/5.2.0
Content-type: text/html

Array
(
    [0] => boolean SearchAlumni(
        string $Name,
        string $Firstname,
        string $Birthdate,
        string $Birthplace,
        string $MatriculationNumber,
        string $NationalInsuranceNumber)
)
```

**Listing 32: Ergebnis der Web-Service-Abfrage**

## 7. Zusammenfassung und Ausblick

Dieses Kapitel fasst die Arbeit zusammen und schließt mit einem Ausblick ab.

### 7.1. Zusammenfassung

Gegenstand dieser Arbeit war die Vernetzung des Alumni-Weiterbildungswbportals mit Diensten des Bildungsportals Thüringen, der Bibliothek und der Alumni Datenbank der TU Ilmenau.

Dazu wurde zuerst das Alumni-Weiterbildungswbportal betrachtet. Es wurde verdeutlicht, aus welcher Situation es entstanden ist und wodurch es sich von anderen Absolventen-Portalen abhebt. Nach den konzeptionellen Darstellungen wurde die konkrete Umsetzung des Portals beleuchtet. Es wurde deutlich gemacht, warum das neue Alumni-Portal nicht durch eine Standard-Portal-Lösung realisiert wird.

Darauf aufbauend wurde sich der Integration von Anwendungen zugewandt. Ein integraler Bestandteil eines Portals ist die Integration. Die von einem Portal angebotenen Dienste und Informationen machen es erst erfolgreich. Aus diesem Grund muss ein Portal Möglichkeiten anbieten, heterogene Anwendungen zu integrieren. Wie erläutert wurde, folgt das neue Ilmenauer Alumni-Portal nicht der Referenzarchitektur und deren Integrationswerkzeuge konnten nicht angewandt werden. Deshalb wurde die Integration allgemeiner betrachtet. Angefangen von Integrationsebenen und -topologien wurde sich den konkreten Integrationstechnologien genähert. Da eine Vielzahl an Technologien existiert, um andere Anwendungen zu integrieren, war es notwendig diese zu klassifizieren. Zudem erleichterte es später die Auswahl einer geeigneten Integrationstechnologie für die zu integrierenden Dienste.

Danach wurde sich den zu integrierenden Diensten zugewandt. Bei der Recherche konnte festgestellt werden, dass das Bildungsportal Thüringen bereits eine Schnittstelle besitzt. Diese wurde analysiert mit der Erkenntnis, dass sie sich schon hervorragend für eine Integration in das Alumni-Weiterbildungswbportal eignet. Eine Neu-Konzeption war deshalb nicht erforderlich. Die beiden Schnittstellen zur Bibliothek und der Alumni-Datenbank der TU Ilmenau mussten hingegen neu konzipiert werden. Es wurde herausgearbeitet, dass die Schnittstelle zur Bibliothek der TU Ilmenau am geeignetsten durch

eine Message-orientierte Middleware, hingegen die Schnittstelle zur Absolventen-Datenbank am besten durch einen Web Service realisiert wird.

Nach der Konzeption der beiden Schnittstellen wurden diese prototypisch implementiert. Die Implementierungen können direkt in die Entwicklung des Alumni-Portals einfließen.

## 7.2. Ausblick

Zum Zeitpunkt der Abgabe dieser Arbeit wird mit Hochdruck an der Fertigstellung des Alumni-Portals gearbeitet. Die Zukunft wird zeigen, ob das Alumni-Portal alle Anforderungen erfüllen und erfolgreich eingesetzt werden kann.

Wichtig für die Zukunft ist das Vorhandensein eines attraktiven Angebots an Informationen und Diensten, denn nur dies garantiert eine dauerhafte Nutzung seitens der Alumni beziehungsweise der wissenschaftlichen Mitarbeiter. Ein Portal lebt von seinen Nutzern. Nur wenn die Nutzer das Portal als Bereicherung ansehen, werden sie es nutzen und eigene Erfahrungen sowie eigenes Wissen einbringen. Dies macht das Portal wiederum attraktiv für andere, wodurch mit der Zeit ein umfangreicher Wissens-Pool entsteht.

Der Erfolg des Alumni-Portals fördert nicht nur die Alumni-Arbeit der TU Ilmenau, sondern steigert auch das Ansehens der Universität gegenüber anderen Hochschulen und der Gesellschaft.

# Anhang

## Anhang A: Analyse der Dienste im Detail

### 1.) *Lebenslauf*

#### *Integrationslokation*

Der Dienst *Lebenslauf* ist einer der wichtigsten Dienste des Alumni-Portals. Ein Portal steht und fällt mit seinen angebotenen Diensten und den damit verbundenen Informationen. Die durch den Dienst Lebenslauf gesammelten Informationen machen das Alumni-Portal für die Nutzer wertvoll. Aus diesem Grund ist der Dienst im Frontend angesiedelt. Ein Nutzer des Portals kann so direkt mit dem Dienst kommunizieren und das Portal mit seinen Informationen füllen.

#### *Informationsfluss*

Der Dienst *Lebenslauf* ermöglicht den Alumni einen Lebenslauf zu pflegen und diese Angaben anderen Nutzern zur Verfügung zu stellen. Damit ist der einzelne Alumni nicht nur Produzent, sondern auch Konsument von Informationen. Legt ein Alumni einen Lebenslauf an, so fließen die Informationen vom jeweiligen Alumni zum Dienst und werden dort gespeichert. Ruft hingegen ein Alumni einen Lebenslauf eines anderen Alumni auf, so fließen die Informationen in die andere Richtung, vom Dienst zum jeweiligen Alumni. Damit liegt ein bidirektionaler Informationsfluss vor.

#### *Kommunikationsmodell*

Der Dienst *Lebenslauf* steht in direkten Kontakt mit den Nutzern des Systems. Ändert ein Alumni die Angaben seines Lebenslaufs oder fordert er einen anderen an, so muss der Dienst sofort antworten, indem er die Informationen aktualisiert oder die angeforderten Daten liefert. Ein Zurückstellen der Anfrage ist nicht sinnvoll. Aus diesem Grund kommuniziert der Nutzer synchron mit dem Dienst.

## Dienstursprung

Der Dienst *Lebenslauf* ist einer der wichtigsten Dienste des Alumni-Portals und deshalb durch das Portal selbst realisiert. Das Alumni-Portal ist Produzent des Dienstes und bietet diesen den Nutzern an.

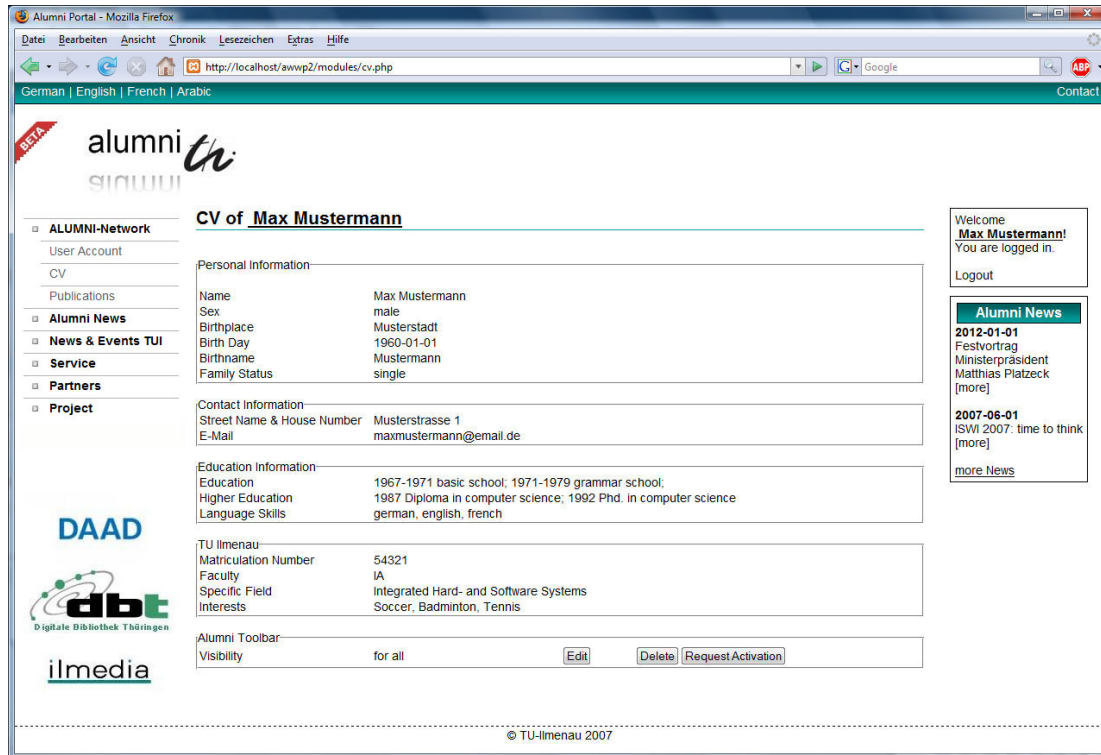


Abbildung 48: Der Dienst Lebenslauf

## 2.) Veröffentlichungen

### Integrationslokation

Der Dienst *Veröffentlichungen* ist ein weiterer wichtiger Dienst des Alumni-Portals. Dieser ist aus denselben Gründen wie der Dienst Lebenslauf im Frontend des Portals zu finden.

### Informationsfluss

Der Dienst *Veröffentlichungen* ermöglicht den Alumni wissenschaftliche Arbeiten zu veröffentlichen. Der einzelne Alumni ist nicht nur Produzent, sondern auch Konsument von Informationen. Legt ein Alumni eine Veröffentlichung an, so fließen die



Informationen vom jeweiligen Alumni zum Dienst und werden dort gespeichert. Ruft hingegen ein Alumni eine Veröffentlichung eines anderen Alumni auf, so fließen die Informationen in die andere Richtung, vom Dienst zum jeweiligen Alumni.

Die Dienste Lebenslauf und Veröffentlichungen sind ähnlich umgesetzt. Beide Dienste sammeln Informationen und stellen diese in einer Wissensdatenbank den anderen Alumni zur Verfügung. Dieser bidirektionale Fluss an Informationen ist typisch für die Dienste einer Web 2.0 Anwendung. Die Dienste einer Web 2.0 Anwendung sind auf die Beteiligung der Nutzer ausgerichtet, die ihre Informationen gegenseitig austauschen und so die Anwendung mit kollektiver Intelligenz anreichern.

#### *Kommunikationsmodell*

Der Dienst *Veröffentlichungen* steht in direkten Kontakt mit den Nutzern des Systems. Ändert ein Alumni die Angaben einer Veröffentlichung oder fordert er eine andere an, so muss der Dienst sofort antworten, indem er die Informationen aktualisiert oder die angeforderten Daten liefert. Wie bei dem Dienst Lebenslauf kommuniziert der Nutzer synchron mit dem Dienst.

#### *Dienstursprung*

Der Dienst *Veröffentlichungen* ist einer der bedeutendsten Dienste des Alumni-Portals und deshalb durch das Portal selbst realisiert. Das Alumni-Portal ist Produzent des Dienstes und bietet diesen den Nutzern an.

Alumni Portal - Mozilla Firefox

http://localhost/awwp2/modules/publication.php

German | English | French | Arabic

alumni *th*

**Publication Input Form (eng)**

ALUMNI-Network

- User Account
- CV
- Publications
- Alumni News
- News & Events TUI
- Service
- Partners
- Project

DAAD

dbt  
Digitale Bibliothek Thüringen

ilmedia

Alumni Author (the email):

E-Mail 1: maxmustermann@email.de

E-Mail 2:

E-Mail 3:

E-Mail 4:

Non Alumni Author (the name):

Name 1: hansmeier@email.de

Name 2: mariamueller@email.de

Name 3: erikamustermann@email.de

Name 4:

Publication Information:

Title: Beispielveröffentlichung

Publish - Date: June 30, 2007

URL: http://www.musterurl.de/...pdf

Abstract: Dieses Paper analysiert...

Keywords:

General field:

Toolbar: [Save] [Reset]

© TU-Ilmenau 2007

Welcome **Max Mustermann!**  
You are logged in.  
Logout

**Alumni News**

2012-01-01  
Festvortrag  
Ministerpräsident  
Matthias Platzeck  
[more]

2007-06-01  
ISWI 2007: time to think  
[more]

more News

Alumni Portal - Mozilla Firefox

http://localhost/awwp2/modules/publication.php?pid=211

German | English | French | Arabic

alumni *th*

**Publication Beispielveröffentlichung (eng)**

ALUMNI-Network

- User Account
- CV
- Publications
- Alumni News
- News & Events TUI
- Service
- Partners
- Project

DAAD

dbt  
Digitale Bibliothek Thüringen

ilmedia

Informations:

Title: Beispielveröffentlichung

Date: 2007-06-30

Authors: Max Mustermann, hansmeier@email.de, mariamueller@email.de, erikamustermann@email.de

URL: http://www.musterurl.de/...pdf

Abstract: Dieses Paper analysiert...

Alumni Toolbar: [Edit] [Delete] [Request Activation]

© TU-Ilmenau 2007

Welcome **Max Mustermann!**  
You are logged in.  
Logout

**Alumni News**

2012-01-01  
Festvortrag  
Ministerpräsident  
Matthias Platzeck  
[more]

2007-06-01  
ISWI 2007: time to think  
[more]

more News

Abbildung 49: Der Dienst Veröffentlichungen

### **3.) Ilmedia**

#### *Integrationslokation*

Da der Dienst *Ilmedia* als Erweiterung des Dienstes *Veröffentlichungen* auftritt, ist dieser dem Backend des Portals zuzuordnen. Er ist vom Nutzer verborgen und wird vom System bedient.

#### *Informationsfluss*

Damit die Publikationen von der Bibliothek katalogisiert werden können, müssen die Informationen der einzelnen Publikationen exportiert und der Bibliothek bereit gestellt werden. Da lediglich die Veröffentlichungen übermittelt werden müssen, findet ein unidirektionaler Informationsfluss vom Dienst *Veröffentlichungen* zum Dienst *Ilmedia* statt.

#### *Kommunikationsmodell*

Die exportierten Veröffentlichungen werden zwischengespeichert, bis die Bibliothek diese bearbeitet. Die Kommunikation zwischen dem Dienst *Veröffentlichungen* und dem Dienst der *Ilmedia* läuft asynchron ab.

#### *Dienstursprung*

Um die Publikationen an die Bibliothek weiterzuleiten, konsumiert das Portal den Dienst *Ilmedia*. Dies ist vollkommen transparent für den Nutzer. Er bekommt davon erst etwas mit, wenn die Bibliothek seine Veröffentlichung geprüft und in ihren Katalog aufgenommen hat.

### **4.) News & Events**

#### *Integrationslokation*

Da der Dienst *News & Events* den Alumni über aktuelle Geschehnisse im Umfeld der TU Ilmenau informiert, ist dieser dem Frontend zuzuordnen.

#### *Informationsfluss*

Der Dienst *News & Events* dient lediglich der Information. Deshalb findet nur ein unidirektionaler Informationsfluss vom Dienst zum Nutzer statt.

### Kommunikationsmodell

Ruft ein Nutzer den Dienst *News & Events* auf, so möchte er die angeforderten Informationen sofort präsentiert bekommen. Deshalb läuft die Kommunikation synchron ab.

### Dienstursprung

Der Dienst *News & Events* ist in das Portal integriert und nicht selbst realisiert. Das Portal tritt als Vermittler ein, der den Dienst unverändert an die Nutzer des Portals weiterleitet.

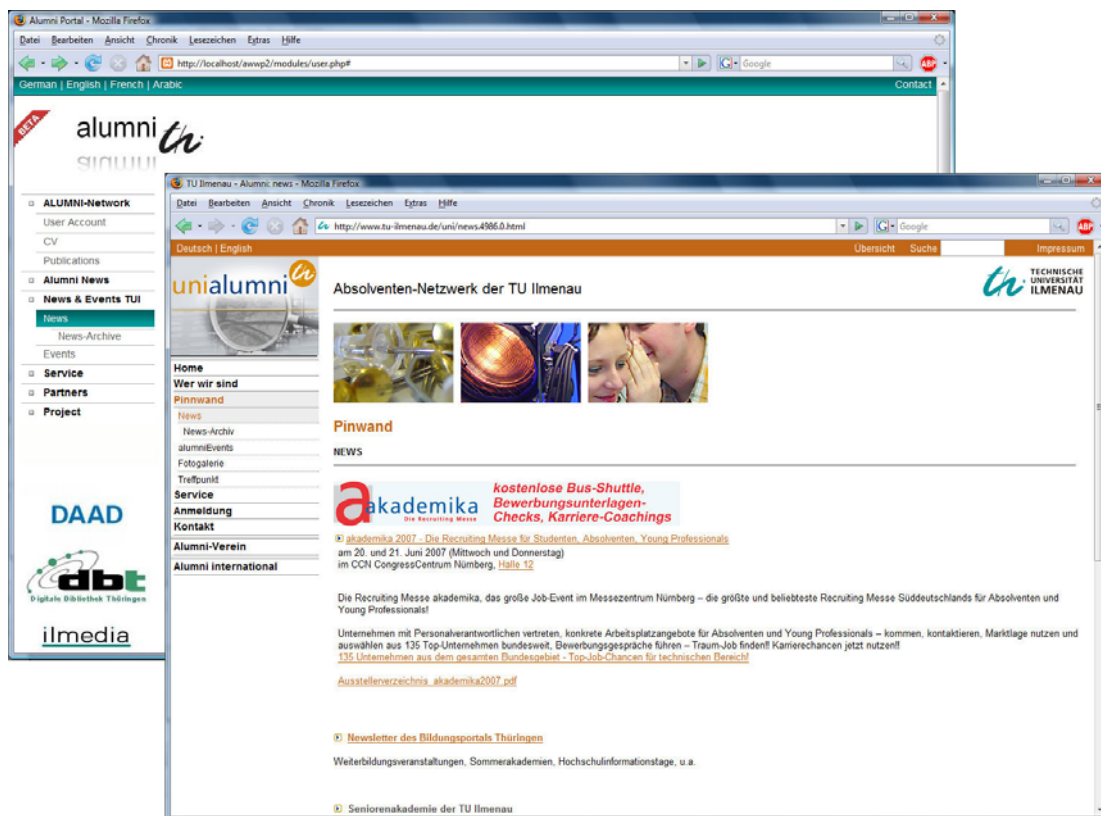


Abbildung 50: Der Dienst News & Events

## 5.) Alumni-News

### Integrationslokation

Da der Dienst *Alumni-News* wie der Dienst *News & Events* den Alumni über aktuelle Geschehnisse informiert, ist dieser Dienst im Frontend zu finden.

## Informationsfluss

Der Dienst *Alumni-News* dient lediglich der Information. Aus diesem Grund findet nur ein unidirektionaler Informationsfluss vom Dienst zum Nutzer statt.

## Kommunikationsmodell

Ruft ein Nutzer den Dienst *Alumni-News* auf, so möchte er die angeforderten Informationen sofort präsentiert bekommen. Daher läuft die Kommunikation synchron ab.

## Dienstursprung

Der Dienst *Alumni-News* ist eine Erweiterung des Dienstes *News & Events*. Da dieser Dienst mehr Informationen enthält, die das Portal betreffen, ist das Portal selbst Produzent des Dienstes.



Abbildung 51: Der Dienst Alumni-News

## **6.) Career-Service**

### *Integrationslokation*

Der Dienst *Career-Service* hat einen rein informativen Charakter und ist deshalb dem Frontend des Portals zuzuordnen.

### *Informationsfluss*

Der Dienst *Career-Service* dient lediglich der Information. Darum findet nur ein unidirektionaler Informationsfluss vom Dienst zum Nutzer statt.

### *Kommunikationsmodell*

Ruft ein Nutzer den Dienst *Career-Service* auf, so möchte er die angeforderten Informationen sofort präsentiert bekommen. Aus diesem Grund läuft die Kommunikation synchron ab.

### *Dienstursprung*

Der Dienst *Career-Service* ist in das Portal integriert und nicht selbst realisiert. Das Portal tritt als Vermittler ein, der den Dienst unverändert an die Nutzer des Portals weiterleitet.



Abbildung 52: Der Dienst Career-Service

## 7.) Jobbörse

### Integrationslokation

Der Dienst *Jobbörse* hat einen rein informativen Charakter und ist deshalb im Frontend des Portals zu finden.

### Informationsfluss

Der Dienst *Jobbörse* dient lediglich der Information. Aus diesem Grund findet nur ein unidirektionaler Informationsfluss vom Dienst zum Nutzer statt.

### Kommunikationsmodell

Ruft ein Nutzer den Dienst *Jobbörse* auf, so möchte er die angeforderten Informationen über die Stellenangebote sofort präsentiert bekommen. Deshalb läuft die Kommunikation synchron ab.

## Dienstursprung

Der Dienst *Jobbörse* ist in das Portal integriert und nicht selbst realisiert. Das Portal tritt als Vermittler ein, der den Dienst unverändert an die Nutzer des Portals weiterleitet.

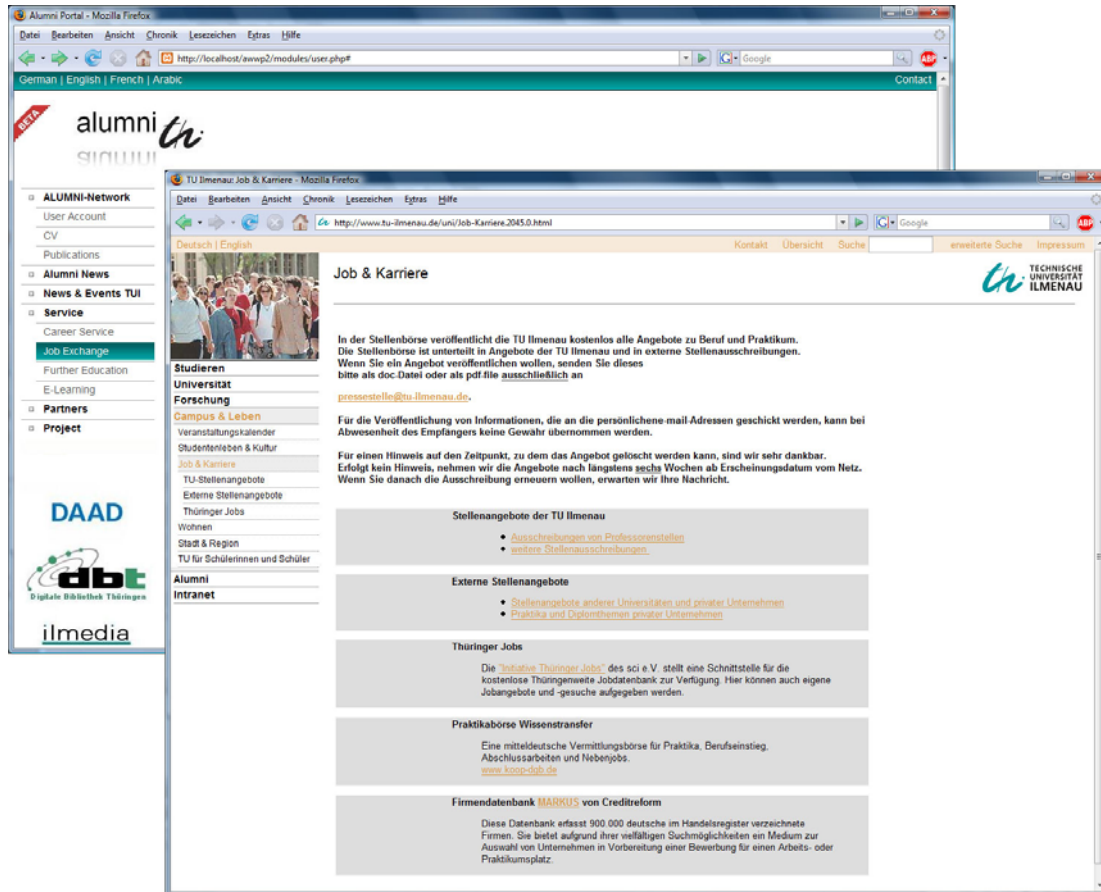


Abbildung 53: Der Dienst Jöbbörse

## 8.) Weiterbildung

### Integrationslokation

Der Dienst *Weiterbildung* hat einen rein informativen Charakter und ist deshalb dem Frontend des Portals zuzuordnen.

### Informationsfluss

Der Dienst *Weiterbildung* dient lediglich der Information. Aus diesem Grund findet nur ein unidirektionaler Informationsfluss vom Dienst zum Nutzer statt.



### Kommunikationsmodell

Ruft ein Nutzer den Dienst *Weiterbildung* auf, so möchte er die angeforderten Informationen sofort präsentiert bekommen. Darum läuft die Kommunikation synchron ab.

### Dienstursprung

Der Dienst *Weiterbildung* ist in das Portal integriert und nicht selbst realisiert. Das Portal tritt als Vermittler ein, der den Dienst unverändert an die Nutzer des Portals weiterleitet.

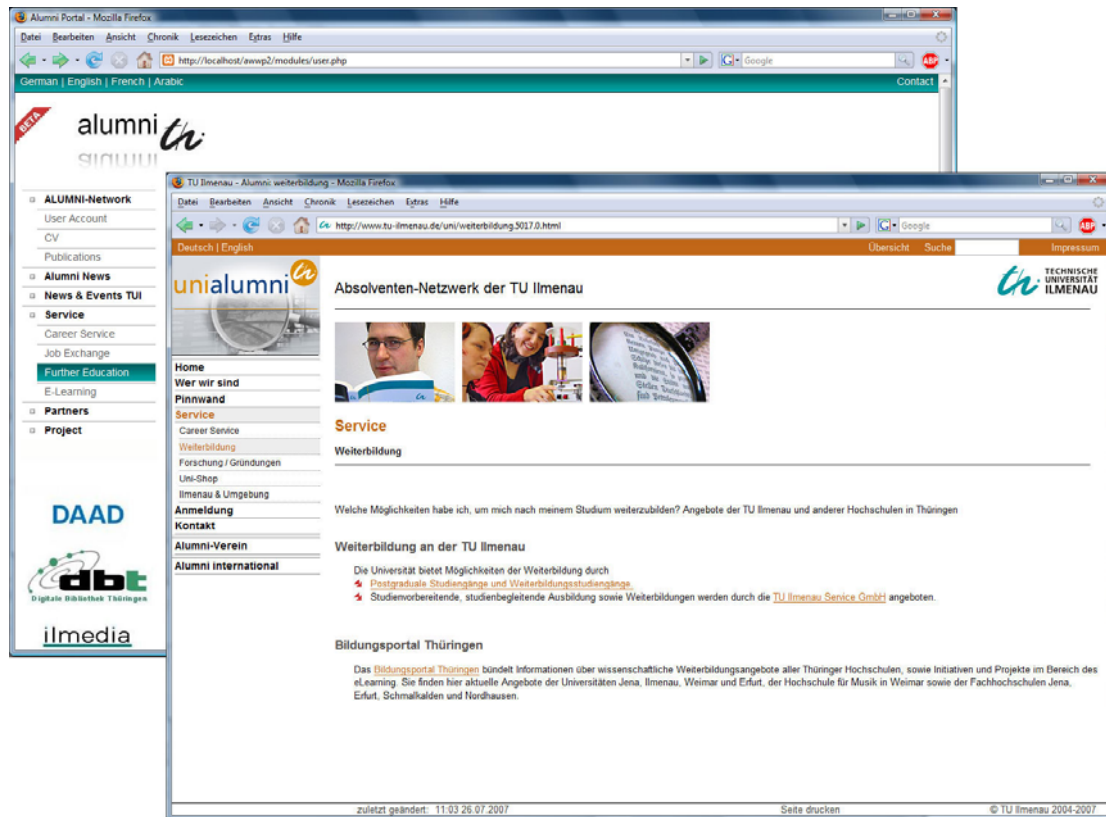


Abbildung 54: Der Dienst Weiterbildung

## 9.) E-Learning

### Integrationslokation

Der Dienst *E-Learning* hat einen rein informativen Charakter und ist deshalb im Frontend des Portals zu finden.

### *Informationsfluss*

Der Dienst *E-Learning* dient lediglich der Information. Deshalb findet nur ein unidirektionaler Informationsfluss vom Dienst zum Nutzer statt.

### *Kommunikationsmodell*

Ruft ein Nutzer den Dienst *E-Learning* auf, so möchte er die angeforderten Informationen sofort präsentiert bekommen. Aus diesem Grund läuft die Kommunikation synchron ab.

### *Dienstursprung*

Im Fall des Dienstes *E-Learning* tritt das Portal sowohl als Vermittler als auch als Produzent auf. Der Dienst nutzt das Bildungsportal Thüringen, um sich über E-Learning-Angebote zu informieren und tritt in diesem Fall als Vermittler auf. Das Portal ist ebenfalls Produzent, indem es eigene E-Learning Materialien anbietet, die die Nutzer des Alumni-Portals in Anspruch nehmen können.

## **10.) Bildungsportal Thüringen**

### *Integrationslokation*

Da der Dienst des *BPT* als Erweiterung der Dienste *Weiterbildung* und *E-Learning* auftritt, ist dieser im Backend angesiedelt. Er ist vom Nutzer verborgen und wird vom System bedient.

### *Informationsfluss*

Der Dienst des *BPT* informiert über wissenschaftliche Weiterbildungsangebote, sowie Initiativen und Projekte im Bereich des E-Learning. Damit von dem Dienst Informationen erhalten werden können, muss eine Anfrage übermittelt werden. Die Antwort enthält dann die gewünschten Informationen. Damit liegt ein bidirektionaler Informationsfluss vor.

### *Kommunikationsmodell*

Da die angeforderten Informationen sofort benötigt werden, läuft die Kommunikation mit dem Dienst synchron ab.

### *Dienstursprung*

Der Dienst des BPT wird vom Portal konsumiert und deren Informationen in die Dienste *Weiterbildung* und *E-Learning* integriert.

### **11.) Präsentation der TU-Ilmenau**

#### *Integrationslokation*

Der Dienst der *Präsentation der TU-Ilmenau* hat einen rein informativen Charakter und ist deshalb im Frontend des Portals zu finden.

#### *Informationsfluss*

Der Dienst der *Präsentation der TU-Ilmenau* dient lediglich der Information. Aus diesem Grund findet nur ein unidirektionaler Informationsfluss vom Dienst zum Nutzer statt.

#### *Kommunikationsmodell*

Ruft ein Nutzer den Dienst *Präsentation der TU-Ilmenau* auf, so möchte er die angeforderten Informationen sofort präsentiert bekommen. Darum läuft die Kommunikation synchron ab.

#### *Dienstursprung*

Der Dienst *Präsentation der TU-Ilmenau* ist eine Sammlung von Informationen für die Präsentation der Technischen Universität Ilmenau. Der Dienst ist durch das Portal selbst realisiert. Das Alumni-Portal ist Produzent des Dienstes und bietet diesen den Nutzern an.

### **12.) Administrator-Funktionalität**

#### *Integrationslokation*

Der Dienst der *Administrator-Funktionalität* ist nur für die Administratoren des Alumni-Portals zugänglich. Zwar ist dieser Dienst den normalen Nutzern des Portals verborgen, trotzdem ist er dem Frontend des Portals zuzuordnen.

#### *Informationsfluss*

Durch den Dienst der *Administrator-Funktionalität* können Administratoren sowohl ihren redaktionellen Aufgaben nachgehen als auch Nutzer benachrichtigen und News veröffentlichen. Da für einen Administrator alle Informationen einsichtig sind, kann er Statistiken aufstellen, diese auswerten und Schlussfolgerungen ziehen.

Die Informationen fließen sowohl vom Administrator, also vom Nutzer des Dienstes zum Dienst, wie auch umgekehrt. Daher liegt ein bidirektionaler Informationsfluss vor.

#### *Kommunikationsmodell*

Ruft ein Nutzer den Dienst *Administrator-Funktionalität* auf, so möchte er die angeforderten Informationen sofort präsentiert bekommen. Aus diesem Grund läuft die Kommunikation synchron ab.

#### *Dienstursprung*

Der Dienst *Administrator-Funktionalität* ist eine Sammlung von Funktionalitäten, die es den Nutzern erlaubt, das Portal zu administrieren. Der Dienst ist durch das Portal selbst realisiert. Das Alumni-Portal ist Produzent des Dienstes und bietet diesen den Administratoren an.

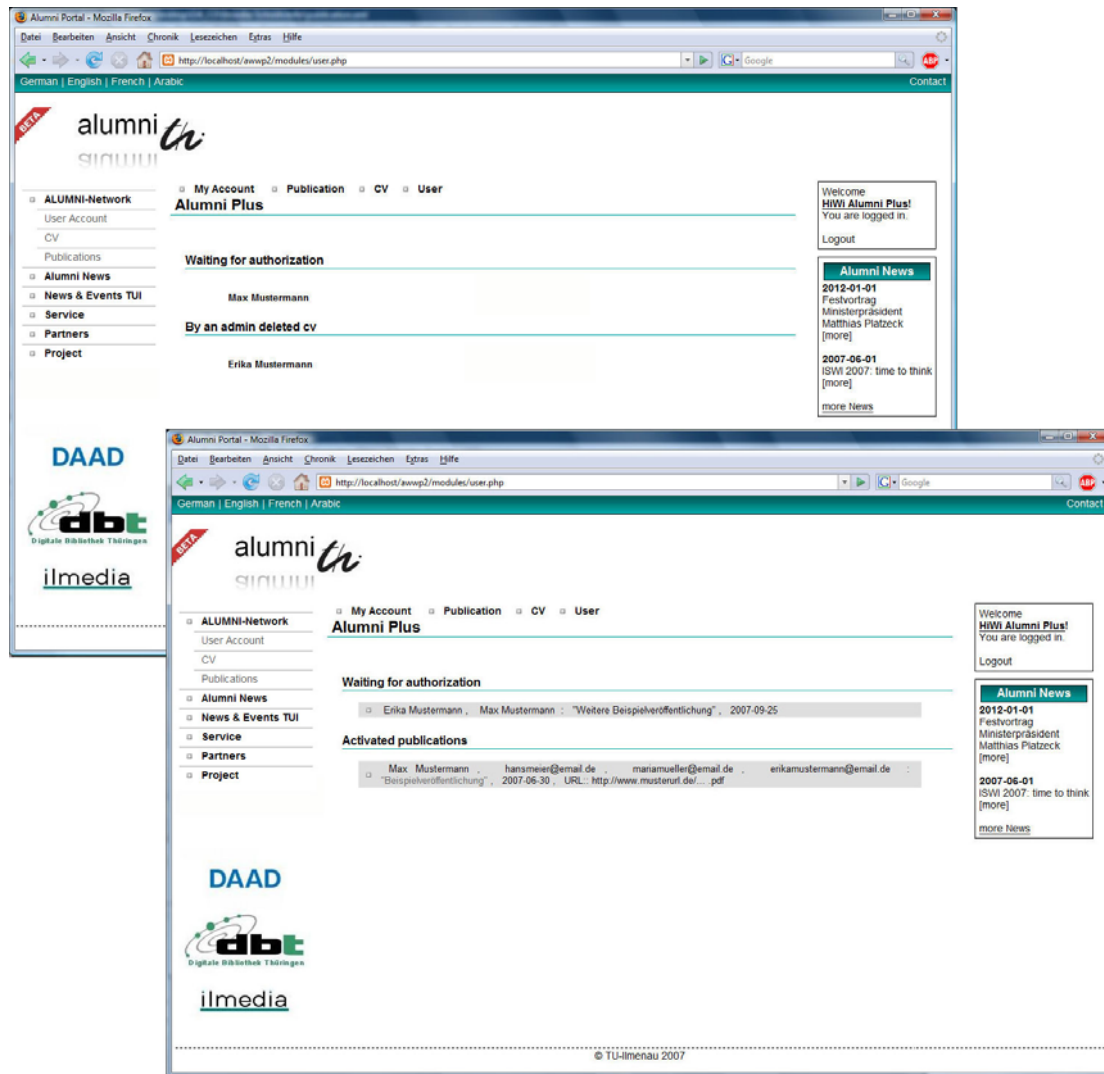


Abbildung 55: Der Dienst Administrator-Funktionalität

### 13.) Nutzerkonto

#### Integrationslokation

Der Dienst *Nutzerkonto* ist ein zentraler Bereich jedes Alumni, indem er seine im Alumni-Portal gemachten Angaben überprüfen kann. Das Nutzerkonto ist dem Frontend des Alumni-Portals zuzuordnen.

#### Informationsfluss

Durch den Dienst *Nutzerkonto* kann jeder Alumni seine gemachten Eingaben bezüglich der Registrierung, der veröffentlichten Lebensläufe und Publikationen überprüfen, ändern und löschen. Die Informationen fließen sowohl vom Nutzer zum Dienst wie auch umgekehrt. Daher liegt ein bidirektionaler Informationsfluss vor.

## Kommunikationsmodell

Ruft ein Nutzer den Dienst *Nutzerkonto* auf, so möchte er die angeforderten Informationen sofort präsentiert bekommen.

## Dienstursprung

Der Dienst *Nutzerkonto* ist durch das Portal selbst realisiert. Das Alumni-Portal ist Produzent des Dienstes und bietet diesen den Nutzern an.

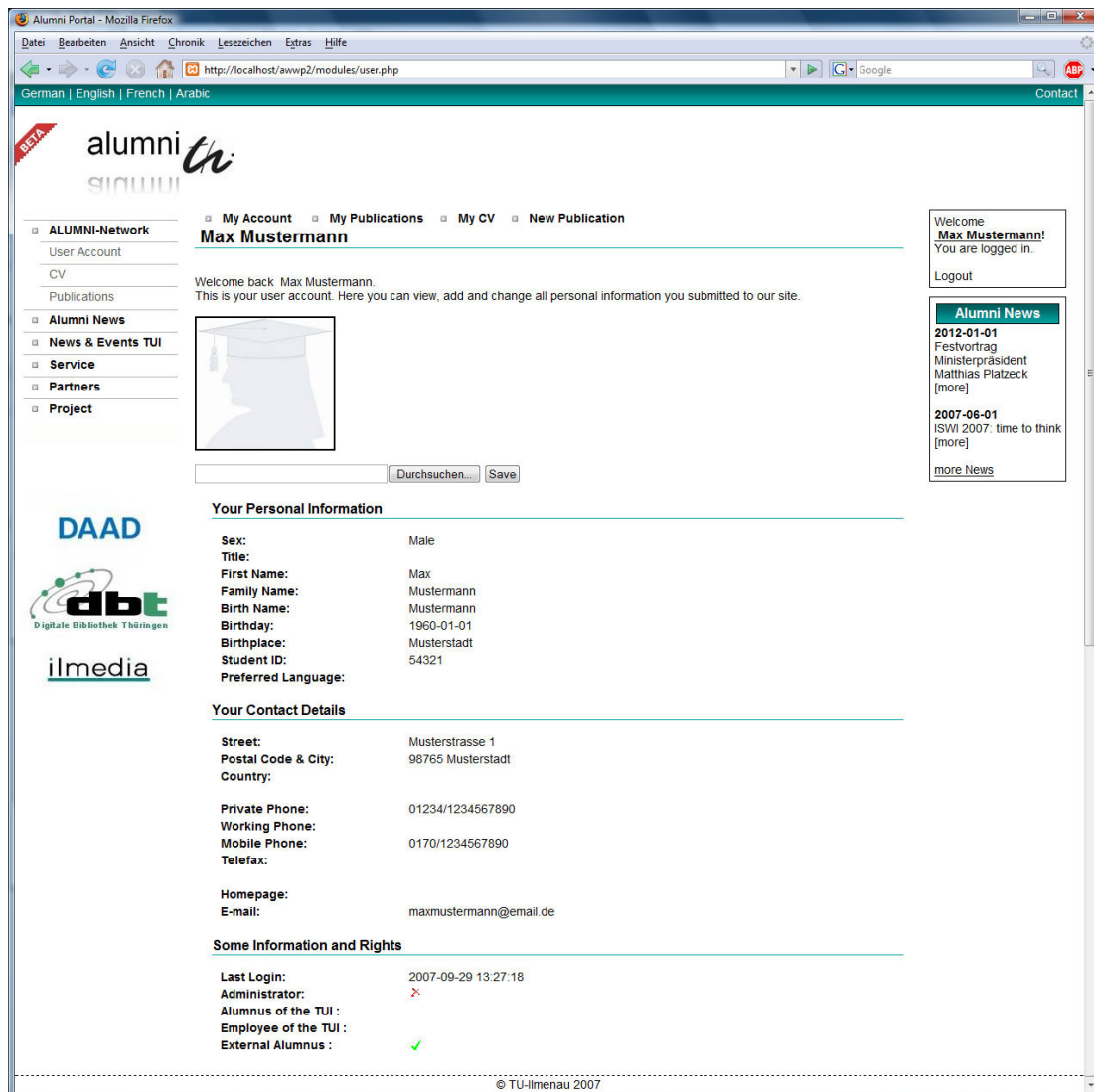


Abbildung 56: Der Dienst Nutzerkonto

## 14.) AlumniDB

### Integrationslokation

Registriert sich ein Nutzer im Alumni-Portal, so wird im Hintergrund durch den Dienst *AlumniDB* geprüft, ob dieser Nutzer ehemaliger Student war oder wissenschaftlicher Mitarbeiter der TU Ilmenau ist. Der Dienst wird vom System bedient und ist deshalb dem Backend zuzuordnen.

### *Informationsfluss*

Damit auf Seite der Alumni-Datenbank geprüft werden kann, ob ein Nutzer Alumni oder wissenschaftlicher Mitarbeiter ist, müssen dem Dienst bestimmte Informationen übermittelt werden. Mit diesen Informationen kann der Dienst eine eindeutige Entscheidung treffen. Das Ergebnis der Überprüfung wird anschließend zurück geschickt. Weil die Informationen in beide Richtungen fließen, liegt ein bidirektionaler Informationsfluss vor.

### *Kommunikationsmodell*

Da die Überprüfung während der Registrierung erfolgt, muss das Ergebnis schnell vorliegen. Ohne eine positive Rückmeldung des Dienstes *AlumniDB* kann eine Registrierung nicht abgeschlossen werden. Aus diesem Grund muss die Kommunikation synchron ablaufen.

### *Dienstursprung*

Der Dienst *AlumniDB* wird vom Portal konsumiert, um den Alumni während der Registrierung zu überprüfen.





# Literaturverzeichnis

- [1] **Gurzki, Thorsten:** *Was ist ein Portal? - Definition und Einsatz von Unternehmens-Portalen*, 2004.
- [2] **Dataquest:** *Portal total*, eNews Magazin, 2002.
- [3] **OvumStudie:** *Enterprise Portals: New Strategies for Information Delivery*, 2000.
- [4] **Schneider, Guido; Zwirger, Florian:** *Sichere Unternehmens-Portale mit SAP*, SAP Press, 2002.
- [5] **Gurzki, Thorsten; Hinderer, Henning:** *Eine Referenzarchitektur für Software zur Realisierung von Unternehmens-Portalen*, erschienen in: *WM2003: Professionelles Wissensmanagement - Erfahrungen und Visionen*, Bonner Köllen Verlag, 2003.
- [6] **Gurzki, Thorsten et al.:** *Marktübersicht Portalsoftware 2005*, Fraunhofer IRB Verlag, 2005.
- [7] **Großmann, Martina; Koschek, Holger:** *Unternehmensportale*, Springer Verlag, 2005.
- [8] **Stelzer, Dirk:** *Portale – Einführung und Überblick*, erschienen in: *Praxishandbuch Portalmanagement*, Gabler, 2004.
- [9] **Meinel, Christoph; Sack, Harald:** *Web-Technologien*, Springer-Verlag, 2004.
- [10] **Badach, Anatol; Rieger, Sebastian; Schmauch, Matthias:** *Web-Technologien – Architekturen, Konzepte, Trends*, Hanser Verlag, 2003.
- [11] **Vonhoeven, Helmut:** *Einstieg in XML*, Galileo Press, 2005.
- [12] **O'Reilly, Tim:** *What is Web 2.0?* [Online] 2005. [Zitat vom: 10. September 2007.] <http://www.oreilly.de/artikel/web20.html>.
- [13] **O'Reilly, Tim:** *Web 2.0: Compact Definition* [Online] 2005. [Zitat vom: 10. September 2007.] [http://radar.oreilly.com/archives/2005/10/web\\_20\\_compact\\_definition.html](http://radar.oreilly.com/archives/2005/10/web_20_compact_definition.html).
- [14] **Bode, Stephan:** *Web Services, Web 2.0 und Semantic Web – Eine Übersicht*. TU Ilmenau, 2006.
- [15] **Alby, Tom:** *Web 2.0. Konzepte, Anwendungen, Technologien*, Hanser Verlag, 2007.
- [16] **Dostal, Wolfgang et al.:** *Service-orientierte Architekturen mit Web Services*, Elsevier Verlag, 2005.
- [17] **W3C: Glossary** [Online] 2004. [Zitat vom: 10. September 2007.] <http://www.w3.org/TR/ws-gloss/>.
- [18] **W3C: Web Service Architectural Roles** [Online] 2002. [Zitat vom: 10. September 2007.] [http://www.w3.org/2002/ws/arch/2/10/roles\\_clean.htm](http://www.w3.org/2002/ws/arch/2/10/roles_clean.htm).
- [19] **Eberhart, Andreas; Fischer, Stefan:** *Web Services*, Hanser-Verlag, 2003.

- [20] **Wöhr, Heiko:** *Web-Technologien*, dpunkt.verlag, 2004.
- [21] **Reenskaug, Trygve:** *Models – Views – Controllers*, 1979.
- [22] **Horn, Thorsten:** *Enterprise Application Integration*. 2004.
- [23] **Böhnke, Jana; Johannes, Hermann:** *ODBC. Optimaler Einsatz im Client/Server-Umfeld*, Addison-Wesley, 1997.
- [24] **Microsoft:** *ADO.NET 2.0*, Microsoft Press Deutschland, 2006.
- [25] **Dicken, Hans:** *JDBC. Web- Datenbank- Integration mit Java*, Redline GmbH, 1997.
- [26] **Völkel, Max:** *Extraktion von XML aus HTML-Seiten*, TH Karlsruhe, 2003.
- [27] **Conrad, Stefan; Hasselbring, Wilhelm; Koschel, Arne:** *Enterprise Application Integration Grundlagen, Konzepte, Entwurfsmuster, Praxisbeispiele*, Spektrum, 2006.
- [28] **Saint-Laurent, Simon et al.:** *Programming Web Services with XML-RPC*, O'Reilly Media, 2001.
- [29] **Stark, Thomas:** *Java EE 5 - Einstieg für Anspruchsvolle*, Addison-Wesley, 2006.
- [30] **Locher, Jens:** *Akzeptanz von Absolventenvereinigungen als Hochschulbindungsinstrument*, Mannheim, 2004.
- [32] **Alumni-Club e.V.** [Online] 2007. [Zitat vom: 10. September 2007.] <http://www.alumni-clubs.net>.
- [32] **Tschöp, Rafaela:** *Alumni-Arbeit an deutschen Hochschulen - Eine empirische Untersuchung im Alumni-Netzwerk der TU Ilmenau*, Diplomarbeit an der TU Ilmenau, 2007.
- [33] **Gartner:** *Magic Quadrant for Horizontal Portal Products*, 2006.
- [34] **Hallbach, Wulf:** *Interfaces : Medien- und kommunikationstheoretische Elemente einer Interface-Theorie*, Fink, 1994.
- [35] **Vogt, Gerhard:** *ilmedia : wissenschaftliches Publizieren an der Technischen Universität Ilmenau*, Universitätsbibliothek Ilmenau, 2005.
- [36] **Reibold, Holger:** *XAMPP kompakt - das offizielle Anwenderhandbuch*, bomots-Verlag, 2005.
- [37] **Bergmann, Sebastian:** *Professionelle Softwareentwicklung mit PHP5*, dpunkt.verlag, 2005.
- [38] **Winkler, Ramona:** *Portals – The All-In-One Web Supersites: Features, Functions, Definitions, Taxonomy*, SAP-AG, 2001.
- [39] **Fricke, Markus:** *Portal-Lexikon der Wirtschaftsinformatik*, Springer, 2001.
- [40] **Gentsch, Peter; Lee, Sue:** *Praxishandbuch Portalmanagement*, Gabler, 2004.
- [41] **Schumacher, Mark; Schwickert, Axel:** *Web-Portale – Stand und Entwicklungstendenzen*, erschienen in: *Arbeitspapiere WI*, Nr. 4/1999, Johannes Gutenberg-Universität, 1999.

- [42] **Ruh, William; Maginnis, Francis; Brown, William:** *Enterprise Application Integration*, Wiley, 2001.
- [43] **Wiehler, Gerhard:** *Mobility, Security und Web Services*, Publicis Corporate Publishing, 2004.
- [44] **Eberhardt, Andreas; Fischer, Stefan:** *Web Services – Grundlagen und praktische Umsetzung mit J2EE und .NET*, Hanser Verlag, 2003.
- [45] **Berners-Lee, Tim; Fischetti, Mark:** *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web*, 1999.
- [46] **Gruber, Thomas:** *A translation approach to portable ontology specifications*, 1993.
- [47] **Dostal, Wolfgang; Jeckle, Mario:** *Semantik, Oden einer Service-orientierten Architektur*, Java Spektrum, 2004.
- [48] **Fensel, Dieter:** *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*, Springer-Verlag, 2003.
- [49] **Richter, Alexander; Koch, Michael:** *Social Software – Status quo und Zukunft*, 2007.
- [50] **Ball, Carson; Evans, Fordin; Haase, Jendrock:** *The Java™ EE 5 Tutorial*, Sun Microsystems, 2006.
- [51] **Boles, Cornelia; Friebe, Jörg; Luhmann, Till:** *Typische Integrationsszenarien und deren Unterstützung durch Web Services und andere Technologien*, EAI Workshop 2004: Enterprise Application Integration, 2004.





# Thesen zur Diplomarbeit

1. Ziel der Arbeit war die Konzeption und Implementierung der Schnittstellen zum Bildungsportal Thüringen, zur Bibliothek und der Alumni-Datenbank der Technischen Universität Ilmenau.
2. Portale bringen Ordnung in die Informationslandschaft, weil sie einen zentralen, einheitlichen und systemunabhängigen Zugang zu Informationen und Diensten schaffen. Sie sorgen für Orientierung in der stetig wachsenden Flut an Informationen.
3. Portale lassen sich klar von einfachen Web-Seiten abgrenzen.
4. Portale sind Web 2.0 Anwendungen.
5. Die Integration von Anwendungen und Daten stellt eine Grundlage für Portale dar.
6. Die meisten Portal-Software-Lösungen am Markt implementieren die Portal-Referenzarchitektur und sind auf Unternehmens-Portale ausgerichtet.
7. Portale eignen sich hervorragend für die Alumni-Arbeit an Hochschulen.
8. Das Alumni-Weiterbildungswbportal fördert die Alumni-Arbeit an der TU Ilmenau und hebt sich von anderen Alumni-Portalen ab.
9. Das Alumni-Weiterbildungswbportal sollte komplett neu entwickelt werden und nicht auf eine Portal-Software-Lösung zurückgreifen.
10. Ein Portal kann einen Dienst vermitteln, konsumieren und produzieren.
11. Es existiert bereits eine Schnittstelle zum Bildungsportals Thüringen und diese eignet sich hervorragend für die Integration in das Alumni-Weiterbildungswbportal.
12. Die geeignetste Schnittstelle zur Bibliothek der TU Ilmenau (Ilmedia) ist durch eine Message-orientierte Middleware realisiert.
13. Die geeignetste Schnittstelle zur Absolventen-Datenbank der TU Ilmenau wird durch einen Web Service realisiert.



# **Eidesstattliche Erklärung**

Ich erkläre an Eides Statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Ich versichere, dass ich dieses Diplomarbeitsthema bisher weder im In- noch im Ausland in irgendeiner Form als Prüfungsarbeit vorgelegt habe.

Ilmenau, den 02.10.2007

---

Christian Saul